The Read Out Controller ASIC for the ATLAS Experiment at LHC

Circuitul integrat ROC pentru experimentul ATLAS de la LHC

Ph.D. Thesis

Transilvania University of Brașov, România

Ph.D. Candidate: Ștefan Popa
Ph.D. Supervisor: Prof. Mihai Ivanovici

Examination Jury:

President:	Conf. dr. ing. Carmen GERIGAN, Transilvania University of Brașov
Coordinator:	Prof. dr. ing. Mihai IVANOVICI, Transilvania University of Brașov
Member:	Senior research fellow Lorne LEVINSON, Weizmann Institute of Science, Rehovot, Israel
Member:	Prof. dr. ing. Gheorghe ȘTEFAN, Politehnica University of Bucharest
Member:	Prof. dr. ing. Liviu GORAȘ, Gheorghe Asachi Technical University, Iași

2021

Abstract

The ATLAS Experiment at LHC is used for fundamental research in particle physics. For its HL-LHC TDAQ system upgrade, new ASICs were developed. The Read-Out Controller (ROC) is such an on-detector radiation-tolerant ASIC that acts as a concentrator, buffer, filter and real-time data packet processor for the new end-cap muon detectors. The thesis presents its elaboration, implementation, quality assurance and control with emphasis on real-world experimental results. The IC is implemented in a 130 nm CMOS technology, resulted in a square die of 22.5 mm^2 with 232 pads and is packaged as 16×16 BGA. The design and its performance model were validated using custom analog and digital functional FPGA-based test setups. The digital test setup emulates the asynchronous chip context, employs optimizations and automatic clock and data synchronization and is used for mass-testing. The ROC's operation was tested while controlled ultrafast neutron beams were incident to its die. Its tolerance to the induced SEUs was evaluated and predictions for the operating environment are made. A proposed implementation of an FPGA Integrated Logic Analyzer that mitigates the observed limitations and constraints of the existing ones is included. The ROC design passed reviews within the ATLAS Collaboration and is included in the TDAQ system.

Rezumat

Experimentul ATLAS de la LHC este utilizat pentru cercetare fundamentală în fizica particulelor. Pentru modernizarea HL-LHC a sistemului său TDAQ, noi ASIC-uri au fost dezvoltate. Circuitul Read-Out Controller (ROC) este un astfel de ASIC tolerant la radiație cu rolul de concentrator, amortizor, filtru și procesor în timp real de pachete de date de la noile detectoare de miuoni. Această teză prezintă elaborarea, implementarea, asigurarea și controlul calității sale cu accent pe rezultate experimentale din lumea reală. Circuitul integrat este implementat într-o tehnologie CMOS de 130 nm, a rezultat într-o pastilă de siliciu pătrată de 22.5 mm^2 cu 232 de pini și este încapsulat ca BGA 16×16. Implementarea și modelul de performanță au fost validate utilizând sisteme de testare funcțională analogică și digitală personalizate și bazate pe FPGA. Sistemul de testare digitală emulează contextul asincron al cipului, conține optimizări și metode de sincronizare automată a semnalelor de date și ceas și este utilizat la testarea în masă. Funcționarea ROC-ului a fost testată sub incidența unor fascicule controlate de neutroni ultra-rapizi. Toleranța sa la efectele imediate de tip SEU ale radiației nucleare a fost evaluată și s-au realizat estimări pentru mediul de operare. O propunere de implementare pentru un Integrated Logic Analyzer pentru FPGA-uri care estompează limitările și constrângerile celor existente este inclusă. Circuitul ROC a trecut cu succes evaluările din comunitatea ATLAS și este inclus în sistemul TDAQ al experimentului.

Acknowledgements

I thank my Ph.D. coordinator, prof. Mihai Ivanovici, for the opportunity to work on a research project related to one of the largest and most prestigious research centers in the world - CERN. Even not considering CERN, he provided a rare opportunity for an electronics engineer, especially one with no experience: contributing to almost all the development stages of an ASIC. He encouraged me to apply to work stages, workshops, winter and summer schools, conferences and competitions. He was also the coordinator for my BSc and MSc diploma projects. He encouraged me to become a research assistant at Transilvania University of Brașov as soon I was eligible. Our collaboration was productive and laid the foundation for future developments within the research center he coordinates. Hopefully, from now on, more ASICs and FPGA-based designs will be developed at the Research and Development Institute of Transilvania University of Brașov with the involvement of students and the help of local industry. Even though I am (still) a dreamer I remain skeptical.

I thank Sorin Mărtoiu from IFIN-HH, Măgurele, România for his insight and help in various specific parts of the ROC-related work, especially the synthesis and implementation steps using the Cadence tools. Without him, the ROC chip would not have been possible. I apologize for the frequent questions and I thank him for his illuminating explanations.

I thank the members of the NSW electronics group at CERN for welcoming a rookie between them and patiently responding to his inquiries.

I thank Radu Coliban for proofreading this thesis and his help in mass-testing the ROC chips at Transilvania University of Brașov.

I thank my parents for supporting and encouraging me during the BSc, MSc and PhD studies. I thank my friends, who also encouraged me and were always interested in my progress.

Contents

Cuprins

Chapter 1

Introduction

The work presented in this thesis relates to the ATLAS (A Toroidal LHC Apparatus) Experiment [103] at the Large Hadron Collider (LHC) [106] particle accelerator, operated by CERN (*Conseil Européen pour la Recherche Nucléaire*, now the *European Organization for Nuclear Research*), situated near Geneva, Switzerland. CERN has in its repertory many important scientific achievements like the discovery of the Higgs boson [99], [100], the production and maintenance of antihydrogen atoms [102] and the birth of the World Wide Web (WWW) information system [148], [4].

In this introductory chapter, all the relevant notions and concepts related to particle physics, the accelerator and the Experiment's purpose and functionality are defined. In the first section, the particle accelerators are described, their purposes and types are presented and their performance measurements are explained. With these notions clarified, the LHC complex is described in the next section. Next, a summary of the current knowledge about the Universe matter and its interactions is presented. Section 1.4 is dedicated to the ATLAS Experiment and its components. The description of the ATLAS Trigger and Data Acquisition (TDAQ) system represents Section 1.5. The planned upgrade of the ATLAS detectors and the associated TDAQ system is detailed in the next section. Section 1.7 depicts the New Small Wheel (NSW) upgrade, part of the ATLAS upgrade, which contains the thesis contributions. In the last two sections, the thesis objectives and the outline of this document are presented.

1.1 Particle accelerators

The idea that every physical object, substance or material from the Universe is composed of indivisible components known as atoms was first proposed by the ancient Greek philosophers (in Greek *atomon* means *uncuttable*). In 1897, J. J. Thomson discovered the electron [113]. The photon (conceptualized by Max Planck in 1900 [168], used in 1905 by Albert Einstein [97] as an explanation for the photoelectric effect[1], nomenclature attributed to Gilbert N. Lewis in 1926 [143]), meson (theorized in 1935 by Hideki Yukawa [218] and discovered in 1947 by Cecil Powell [139]) and other particles followed in a relatively short time. Therefore, one could consider that particle physics was born when the electron was discovered. The aim has been to explain the structures of matter and their interactions as accurately as possible

[1]Electrons are released by a material when it absorbs electromagnetic radiation.

[67]. The atom proved to be divisible, consisting of a nucleus, representing more than 99.94 % of its mass and one or more electrons. The nucleus was then proved to be a group of one or more protons and some neutrons (i.e. only the most common hydrogen nucleus contains no neutrons). The notion of elementary or fundamental particle was introduced to denote the subatomic particles with no substructure [67]. At first, the neutrons and protons were considered fundamental particles but subsequently it was found that they contain quarks which at the moment of writing this thesis are considered fundamental. The Standard Model (SM) of particle physics is the theory that classifies all known elementary particles and describes three out of the four known fundamental forces [121], [67].

Each theoretical hypothesis must be proved right or wrong by repeatable and objective experimentation. Thus, to validate the theoretical assumptions related to the atomic nucleus, particle accelerators [11] were built since 1930 - 1932 [77], [78] (i.e. the first artificial nuclear disintegration achieved by Cockroft and Walton by colliding accelerated protons into lithium). A particle accelerator is a system that forms and transfers energy into well-defined beams of particles using electromagnetic fields. The electrical fields provide the acceleration while the magnetic fields concentrate and direct the beam. Thus, the particles contained in the beam reach very high speeds and are focused as much as possible. In the early stages of development and usage, the target of the particle beam was a piece of material since the goal was to investigate the structure of the nuclei. That is why in the 20^{th} century the particle accelerators were named *atomic smashers* [88], [9], [22]. The majority of modern accelerators collide directly subatomic particles [69].

Besides fundamental research, particle accelerators have many other uses in technical and industrial fields: e.g. particle therapy (treatment of cancer), radiation sterilization of medical devices, ion implantation (semiconductor device fabrication), nuclear physics (production of isotopes), etc. By the method of transferring energy to the beam, particle accelerators are divided into two classes: electrostatic accelerators which use static electric fields (e.g. Van de Graaff generators) and electrodynamic (also named electromagnetic) accelerators which employ varying electromagnetic fields and can reach higher energies (e.g. modern large-scale accelerators like the LHC). There are two constructive types of electrodynamic particle accelerators depending on the beam trajectory: linear and circular [69].

The performance of a particle accelerator is determined using two metrics: the energy transferred to the particle beam and the luminosity [128]. The higher the energy of the colliding particles is, the higher the probability of generating a particle with a higher mass and the possibility of reaching further into the structure of matter [121]. This energy refers to the kinetic energy of the particles gained in the accelerator. The usual unit of measure for the energy of a particle is the electronvolt (eV) and its multiples keV (10^3 eV), MeV (10^6 eV), GeV (10^9 eV), TeV (10^{12} eV), etc. One eV is the amount of energy gained by an electron accelerated by a 1 V potential difference (i.e. $1\text{ eV} = 1.602176634 \times 10^{-19}\text{ C} \times 1\text{ V} = 1.602176634 \times 10^{-19}$ J). As a reference, the LHC collides proton beams of 7 TeV each, resulting in a total collision energy of 14 TeV. By converting this energy into Joules and comparing it with the kinetic energy gained by an object of 1 kg mass falling from a height of 1 m, as detailed below in equations 1.1 and 1.2, one can conclude that this is a very small amount of energy. But considering the mass of one particle from the beam (e.g. $m_{\text{proton}} = 1.67262192369(51) \times 10^{-27}$ kg), this energy is sufficient to accelerate the

particle to 99.9999991 % of the speed of light in vacuum ($c = 299,792,458\, m \cdot s^{-1}$). No particle can travel faster than c but there is no energy threshold. Through Einstein's famous mass and energy equivalence equation (i.e. $E = mc^2$), the mass of the accelerated particle increases with the rise of energy.

$$E_{LHC} = 14 \times 10^{12} \times 1.602176634 \times 10^{-19} = 22.4 \times 10^{-7} J \tag{1.1}$$

$$E_{\text{Newtonian kinetic}} = \frac{1}{2} \cdot mv^2 = E_{\text{potential}} = mgh = 9.8J,\ g = 9.8\, m \cdot s^{-2} \tag{1.2}$$

The luminosity (L) is a performance metric for particle accelerators defined as the ratio between the number of particle interactions (R) produced in a set time (t) and the cross-section (σ) of the interaction [128]:

$$L = \frac{1}{\sigma} \frac{dR}{dt} \tag{1.3}$$

In physics, the cross-section σ represents the probability that a specific event will occur when a radiant phenomenon intersects a localized object or variation of density. It has the same unit of measure as the surface area since it represents the transverse size of the targeted object that the radiant phenomenon must hit for the process to take place. Thus, the unit of measure for the luminosity is $\text{cm}^{-2} \cdot \text{s}^{-1}$.

The collisions of the accelerated particles can be with a fixed target or with another incoming accelerated beam (like within the LHC). In the latter case, both beams are considered simultaneous targets and incoming beams and the resulted luminosity depends on the beam size and the number of particles per bunch (if the beam is not a continuous flux of particles) [128], as detailed in equation 1.4 which assumes bunches with equal Gaussian[2] profiles (i.e. densities) in the x-y plane perpendicular to the direction of the beams (i.e. $\mu_{1x} = \mu_{2x} = 0$, $\sigma_{1x} = \sigma_{2x} = \sigma_x$, $\mu_{1y} = \mu_{2y} = 0$, $\sigma_{1y} = \sigma_{2y} = \sigma_y$, where σ is the standard deviation and μ the mean of the normal distributions) and symmetric Gaussian profiles on the z axis relative to the central collision point (i.e. $\mu_{1z} = -\mu_{2z} = s_0$, $\sigma_{1z} = \sigma_{2z} = \sigma_z$, where s_0 is the modulus of the distance between the bunches center of mass and the central collision point). N_1 and N_2 are the numbers of particles contained in the two colliding bunches, N_b represents the number of bunches in the beam and f_{rev} is their rate of revolution. The x, y and z axes are considered the same as for the ATLAS Experiment, explained in Section 1.4. One can observe that the luminosity does not depend on the bunches length σ_z.

$$L = \frac{N_1 N_2 f_{rev} N_b}{4\pi \sigma_x \sigma_y} \tag{1.4}$$

Because the luminosity can vary in time, the final figure that reflects the number of observed events and as a result the quantity of produced data is the integrated

[2]Normal probability density function $\rho(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$ with mean μ and variance σ^2.

luminosity (the differential $dt^{'}$ indicates that the variable of integration is $t^{'}$ which is the infinitesimal of t from eq. 1.3):

$$L_{int} = \int_0^T L(t^{'})dt^{'}$$

The integrated luminosity excludes the periods when the machine is not operational. It represents the total data quantity over the sensitive time related to an experiment. One unit of measure for the area of a surface is the femtobarn, fb, where *femto* is the prefix meaning a factor of 10^{-15} and *barn* equals 10^{-28} m^2. Thus $1fb = 10^{-43}$ m^2. As a reference, the LHC reached a peak instantaneous luminosity of $L = 2 \times 10^{34}\text{cm}^{-2} \cdot \text{s}^{-1}$ for the proton collisions in 2018 and accumulated 160 fb^{-1} of proton collisions data between 2015 and 2018 [198].

1.2 The ATLAS-LHC-CERN context

The LHC is the largest and highest-energy artificial synchrotron in the world [106]. A synchrotron is a type of electrodynamic circular particle accelerator in which the particle beam follows a closed-loop trajectory and the intensity of the magnetic field that bends the beam on its path is synchronous to the beam's energy. The word *artificial* is important for accuracy when talking about LHC since the highest-energy particle accelerators in the Universe are black holes.

LHC is situated in an underground tunnel, as deep as 170 m from the ground level and has a circumference of 27 km [106], beneath the border of Switzerland and France, near Geneva. It was built between 1998 and 2008 and is operated by CERN. Within the accelerator, proton (with notation $^1_1\text{H}^+$, p, p^+ or N^+) or ion beams travel in opposite directions. The $^1_1\text{H}^+$ beams are organized in bunches of roughly $1.15 \cdot 10^{11}$ particles (i.e. at the start of beam run) and are spaced at 7.5 m, corresponding to a travel time of approx. 25 ns [177]. The energy reaches 7 TeV per proton. The beams' paths intersect each other in four collision points. Detectors that provide the necessary signals for the determination of the trajectory, energy and electrical charge of the resulted particles from the collisions are installed in the four locations: ALICE (A Large Ion Collider Experiment) [101], ATLAS [103], CMS (Compact Muon Solenoid) [104] and LHCb (LHC beauty) [105].

The structure of the LHC accelerator is depicted in Figure 1.1, with the acronyms detailed in Table 1.1. The different types of particles are depicted with arrowheads of distinct colors. Some facilities are labeled with the year of commissioning while the synchrotrons are also labeled with their circumferences in brackets. The following descriptions refer to the production and acceleration of proton beams.

Hydrogen atoms are ionized in an electric field (the electrons are stripped from the proton-only nuclei resulting in protons $^1_1\text{H}^+$) and inserted into a series of particle accelerators to reach higher and higher energy levels. These accelerators are LINAC 2 (LINear ACcelerator Generation 2) which outputs protons with 50 MeV energy, PSB (Proton Synchrotron Booster) formed by four superimposed synchrotron rings which further increase the beam energy to 1.4 GeV, PS (Proton Synchrotron) which produces beams with energies up to 25 GeV, SPS (Super Proton Synchroton) which further increases the energy up to 450 GeV, currently being the second-largest par-

ticle accelerator in CERN's complex and the LHC. The proposed FCC (Future Circular Collider) extends the series, having a circumference of 100 km and reaching a 100 TeV proton-proton collision energy [37].

Acronym	**Description**
LINAC 1, 2, 3, 4	LINear ACcelerator Generation 1 [127] (commissioned in 1959, it produced and accelerated $^{1}_{1}H^{+}$ up to 50 MeV, resulting in a beam current of maximum 70 mA, operational until 1992), 2 [12] [13] (commissioned in 1978, it produced and accelerated $^{1}_{1}H^{+}$ up to 50 MeV, but reached a maximum 150 mA beam current, operational until 2018), 3 [14] (commissioned in 1994 in the place of LINAC 1, it produces and accelerates heavy ions for the PSB and is still in use) and 4 [144] (currently being tested after commissioning, it replaces LINAC 2 for $^{1}_{1}H^{+}$ beams and reaches 160 MeV).
LEIR	Low Energy Ion Ring [59] transforms the long heavy-ion pulses from LINAC 3 into shorter but denser bunches for PS.
PSB	Proton Synchrotron Booster [126] (BOOSTER) is the smallest synchrotron at CERN, it accelerates the protons from LINAC 2/4 up to 1.4 GeV.
ISOLDE	Isotope Separator On Line DEvice [10] [73] is a facility that produces and studies RIBs, proposed in 1964, first started in 1967 and still in operation.
REX-ISOLDE	Radioactive beam Experiment at ISOLDE [125] is an experiment at ISOLDE for testing new concepts related to RIBs.
HIE-ISOLDE	High Intensity and Energy ISOLDE [65] is a major upgrade of ISOLDE aiming at improving the RIB's quality and energy.
RIB	Radioactive Ion Beam. Beam of unstable (i.e. radioactive) nuclides (atoms with a specific number of neutrons and protons in the nucleus). E.g. produced within ISOLDE.
PS	Proton Synchrotron [70], the first synchrotron from CERN (1959), accelerates the protons from PSB or the heavy ions from LEIR up to 25 GeV and 72 MeV, respectively.
CLEAR	CERN Linear Electron Accelerator for Research [115], is a general-purpose research facility exploring different accelerator technologies.
n-ToF	neutron Time-Of-Flight [63] [87] facility is a neutron source with energies ranging from a few meV up to GeV.
AD	Antiproton Decelerator [58] [187] is a low-energy antiproton beam facility used for producing and studying antimatter.
ELENA	Extra Low ENergy Antiproton [145] is a storage ring within AD that decelerates the antiproton beam to 0.1 MeV.
SPS	Super Proton Synchroton [8] is a synchrotron with a circumference of 6.9 km that accelerates the particles (i.e. protons and heavy ions) from PS up to the equivalent 450 GeV proton energy.
AWAKE	Advanced Proton Driven Plasma Wakefield Acceleration Experiment [124] represents the proof of principle experiment for a new type of accelerator that uses plasma.
HiRadMat	High-Radiation to Materials [76] is a facility that provides high-intensity, high-energy proton and ion beams for material and component testing.
TT2/10/20/40/ 41/42/60/66, TI2/TI8	Transfer Tunnels or Transfer Lines [56], deliver accelerated particles between synchrotrons or between synchrotrons and the experiments, e.g. TT2 delivers the products of the PS to n-ToF, AD and SPS.

Table 1.1: The descriptions of all the acronyms from Figure 1.1.

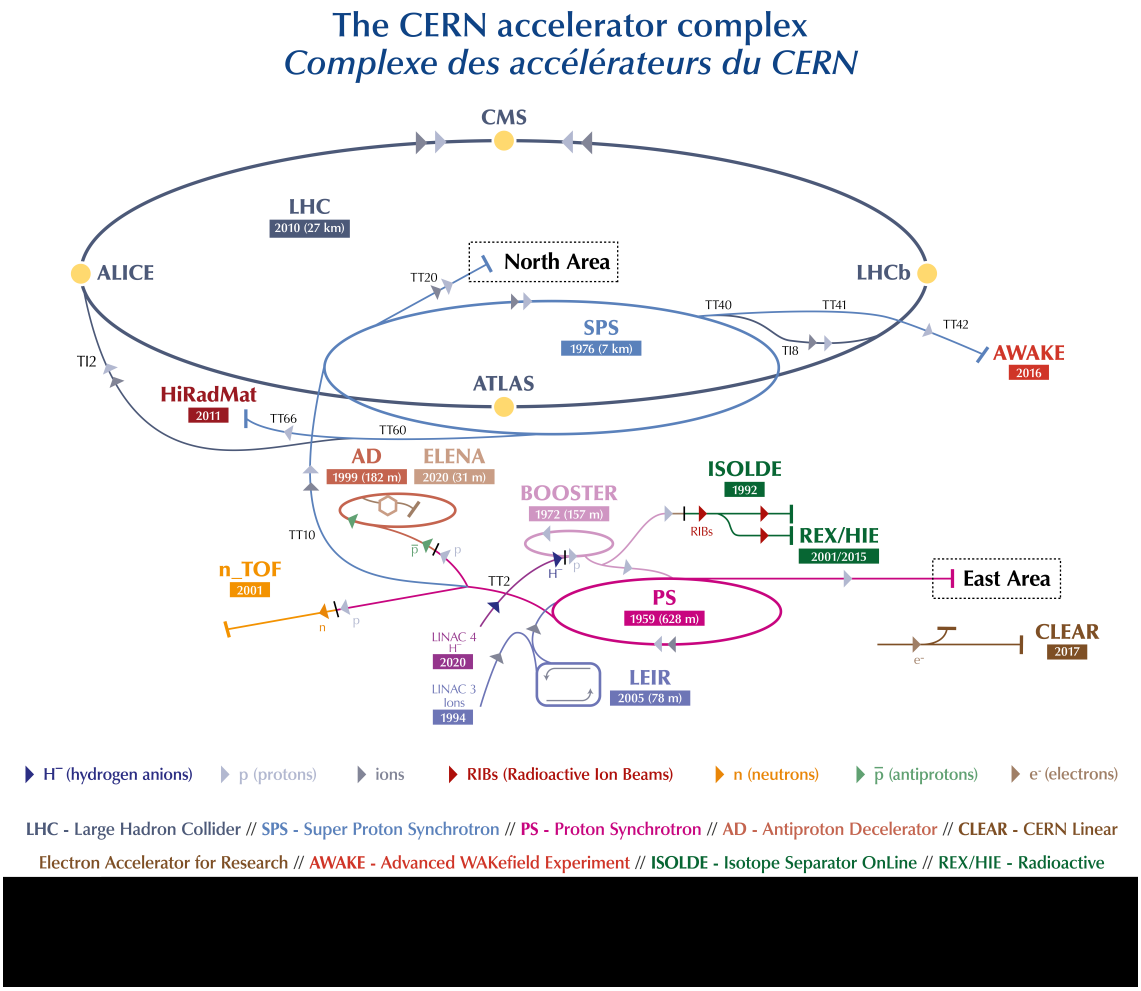


Figure 1.1: The LHC particle accelerator system [5].

The accelerated protons within the LHC's tunnel are clustered into up to 2808 bunches distributed along the circumference [106] [177]. Each one contains up to 100 billion particles and has variable size depending upon the location. When they are far from any interaction points, they are a few cm long and one mm wide [75]. The minimum size (i.e. 20 μm) is obtained around the interaction points to increase the collision probability [75]. Up to 40 collisions are produced in each Bunch Crossing (BC) [75]. Even if the bunches are spaced at 25 ns, corresponding to a BC rate of 40 MHz, due to practical reasons there are gaps in their pattern so, in one second, an average of 3×10^7 bunches cross [75] [177]. This amounts to 10^9 collisions every second [74] [75] from which several hundreds are of interest. The bunches travel like this within the tunnel between 10 and 24 hours [75]. New bunches are then formed and accelerated.

From the four LHC experiments, ATLAS and CMS are general purpose and have high luminosity, while ALICE is dedicated to the physics of heavy ions and LHCb is aimed at studying the bottom quark.

1.3 The Standard Model of particle physics

The classification of all known elementary particles and the description of three of the four known fundamental forces in the Universe form the SM theory of particle physics [121]. It has been developed in the second half of the 20$^{\text{th}}$ century [159]. Although this theory demonstrated correct experimental predictions, it is still not complete and does not fully explain some phenomena like the matter-antimatter asymmetry [71] and the dark matter and energy [72]. In Figure 1.2 the 17 fundamental particles (12 fermions and 5 bosons) as of 2019 are presented. Certain masses are periodically redetermined by the scientific community. Through the mass-energy equivalence (i.e. $E = mc^2$), the masses of the fundamental particles are expressed in electronvolts over c^2. As a reference, the equivalent mass of 1 eV/c^2 is:

$$1\ \text{eV}/c^2 = \frac{(1.602176634 \times 10^{-19}\text{C}) \times 1\text{V}}{(2.99792458 \times 10^{8}\text{m/s})^2} = 1.7866192 \times 10^{-36}\text{kg}$$

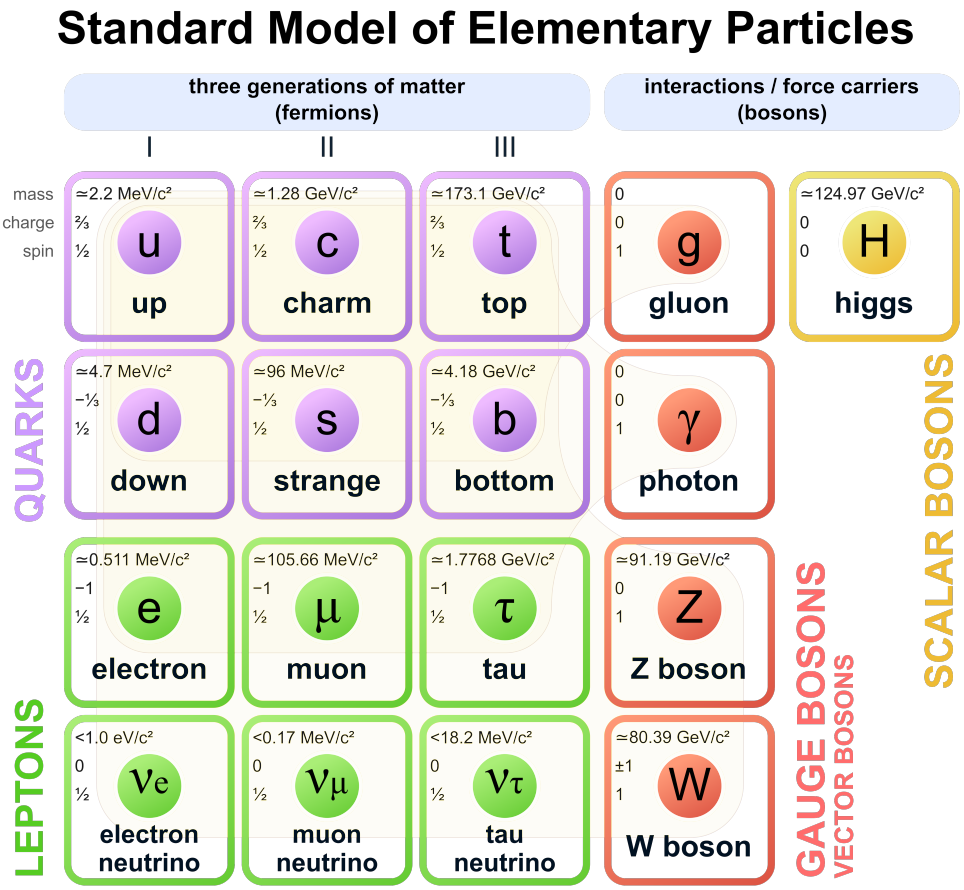


Figure 1.2: The 17 elementary particles of the SM of particle physics [152].

The spin of an elementary particle is an intrinsic property inferred from experiments and theorems equivalent to a form of angular momentum. It is usually expressed as spin quantum numbers (like in Figure 1.2) and can have half-integer values (spin quantum number $s = \frac{n}{2}$, where n is any non-negative integer). The spin quantum number depends solely on the type of particle and cannot be changed.

The SM describes 12 elementary particles, all with the spin quantum number equal to 1/2, named fermions, that form the matter in the Universe. For each fermion type, an associated particle exists having the same mass but opposite physical charges (e.g. electrical charge) called antiparticle (e.g. the antiparticle of the electron is the positron). Based on the electrical charge, the fermions are divided into six leptons (which have an integer electrical charge value) and six quarks (with a non-integer electrical charge value). Both categories are divided into pairs called generations (see Figure 1.2) based on similar physical behavior. The generations are numbered such that the corresponding members of the lower generation have a lower mass. From the lepton category, the electrons are the most common in the Universe since they are stable. Muons and Taus are produced only in high-energy collisions and decay into electrons and neutrinos. The quarks exist only in combinations called hadrons (e.g. protons and neutrons) and are the only fundamental particles that participate in all four fundamental interactions.

The fundamental interactions, also called fundamental forces, are the interactions between particles that are currently considered the most basic. There are two known nuclear fundamental interactions (the strong and weak interactions) that have subatomic ranges and two extra-nuclear fundamental interactions (gravitational and electromagnetic) which cause long-range forces that extend outside the atomic nucleus. The stability of matter is maintained by the strong interaction (constraining the quarks into hadrons, linking neutrons and protons into atomic nuclei) while the weak interaction produces the radioactive decay of atoms.

Except for gravity, the SM assumes that all interactions are mediated by force carriers (exchange particles) called gauge bosons. The massless photon is responsible for the electromagnetic interaction. The relatively massive W^+, W^- and Z bosons are responsible for the weak interactions and the eight massless gluons mediate the strong interactions between quarks. The massive Higgs boson, theorized in 1964 and independently found by the ATLAS and CMS experiments from LHC in 2012, has the role of generating the masses of fermions. As a reference, the production rate within LHC for the Higgs boson is about one in a billion collisions (the collision rate is up to one billion per second, thus the Higgs rate is up to one per second) [75]. The boson decays very fast so its identification and measurements are achieved using its decay products. This means that a very small number of them are detectable.

For the special case of gravity, the SM predicts the graviton as a mediating elementary particle but it is undiscovered as of 2021. Currently, gravitation is best understood in Einstein's general theory of relativity [96] which was confirmed at large scale. In this theory, gravitation is determined by the geometry of spacetime.

New theories were created that consider other types of particles [192] to complete the SM. One important example is supersymmetry (SUSY) which incorporates the gravity and proposes new particles such as sfermions and gaugino [95].

1.4 The ATLAS detector

The ATLAS detector, depicted in Figure 1.3, has a cylindrical shape around the interaction point of the Experiment, orientated so that the two incoming particle beams are perpendicular to the bases in their centers. The height, equal to the diameter of the cylinder base, is 25 m and the length, the distance between the bases, is 44 m. It is designed to be general-purpose, i.e. not focusing on a specific physical

phenomenon and the associated signals but maximizing the range of measurable signals. It is intended to characterize any particles produced from the beams collision (i.e. determine their masses, momentum, lifetimes, charges, spins and energies). To achieve this requirement it contains different types of detectors, organized in layers of cylindrical shape, called *barrel* sections. The disk detector assemblies, parallel to cylinder bases, which close a barrel section are referred to as *end-cap* sections. The detector has no blind spots, being hermetic. The higher the energy of the particles is, the larger the detector must be to effectively characterize the resulted products by stopping them. Neutrinos are the only stable particles that are not directly detected but inferred from the inequality of the momentum of the detected particles [179].

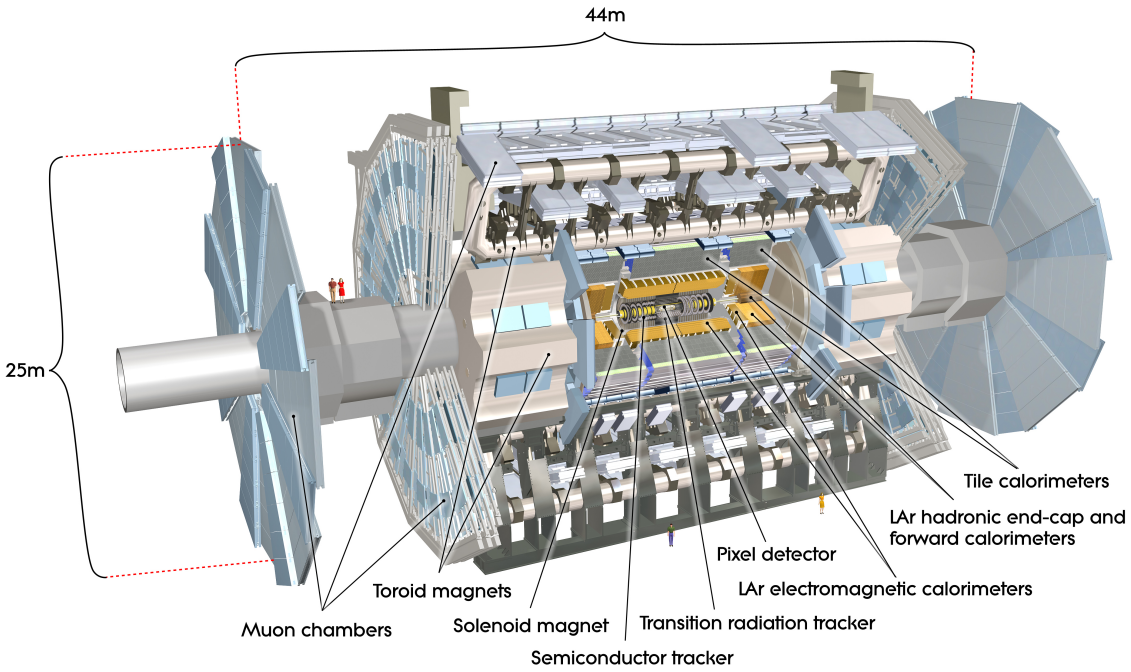
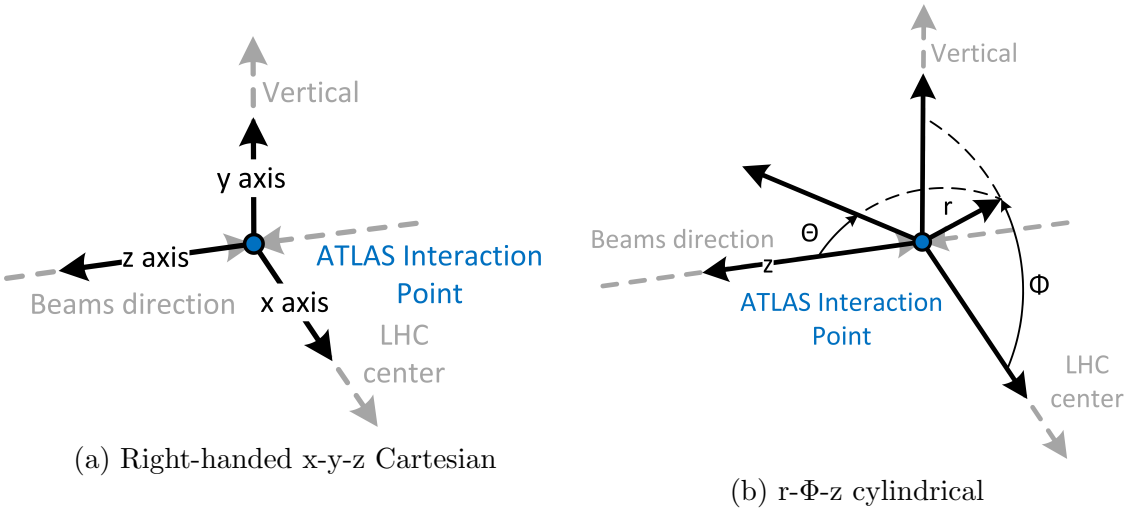


Figure 1.3: The ATLAS detector [165].

It is essential to understand the system of coordinates that ATLAS uses to describe its structures and functionalities. This system is three-dimensional right-handed Cartesian with the origin set in the interaction point and the z axis indicating the direction of the beams [103], as depicted in Figure 1.4a where it is considered that the ATLAS detector has the same orientation as in Figure 1.3. The x-y plane is thus perpendicular to the beams, parallel to the cylinder bases [103]. The y axis points upwards and the x-axis to the center of the circle described by LHC [103]. The equivalent cylindrical coordinates, depicted in Figure 1.4b, keep the z coordinate but express the position in the x-y plane as r-Φ, where r is the distance from z axis and Φ is the azimuthal angle ($0 \leq \Phi \leq 2\pi$) [103]. Often, the trajectories of particles produced in the interaction point of the Experiment are expressed by two angles: the same azimuthal angle Φ used in the cylindrical coordinate system and the polar angle Θ ($0 \leq \Theta \leq \pi$), also depicted in Figure 1.4b, as the angle from the z axis in the r-z plane [103]. As a substitute for the polar angle, the rapidity and its approximation at high energies, the pseudorapidity η, describe the regions covered by the detector's components. The transformation of the polar angle into

pseudorapidity η is as follows [103]:

$$\eta = -\ln \tan \frac{\Theta}{2}$$



(a) Right-handed x-y-z Cartesian

(b) r-Φ-z cylindrical

Figure 1.4: The two coordinate systems used to describe the locations within the ATLAS Experiment. The ATLAS detector has the same orientation as depicted in Figure 1.3. Inspired by [167].

The main constituents of the ATLAS detector are the Inner Detector (IDET), the calorimeters, the Muon Spectrometer (MS) and the magnet system. All these sub-systems are divided into multiple layers: the IDET, detailed in Section 1.4.1, contains the Pixel detector, Semiconductor tracker and the Transition Radiation Tracker; the calorimeters, detailed in Section 1.4.2, are represented by the Liquid Argon electromagnetic calorimeters, the Liquid Argon hadronic end-cap and forward calorimeters and the Tile calorimeters; the magnet system is comprised of the Toroid Magnets and the Solenoid Magnet and the MS, detailed in Section 1.4.3, contains Thin Gap Chambers (TGCs), Resistive Plate Chambers (RPCs), Monitored Drift Tubes (MDTs), and Cathode Strip Chambers (CSCs).

The sub-detectors are complementary: the IDETs determine the trajectory while the calorimeters measure the energy of the particles that are stopped within them. The highly penetrating muons are measured by the muon system (i.e. both tracking and energy measurement). The tracking system is sensible to charged particles (i.e. muons, charged hadrons and electrons) while the calorimeters interact with both the charged and neutral particles and resulting in particle showers [188]. These interactions are depicted in Figure 1.5. The neutrinos do not interact at all with the entire detector [188]. The muons do not cause particle showers but ionize [188]. Photons and electrons produce particle showers over relatively short distances while the hadrons penetrate further into the detector and produce larger showers [188].

In addition to the different types of sub-detectors, a magnet system complements the system [18]. The secondary particles resulted from the particle collisions in the interaction point, fly out in all directions. To measure the momentum of the resulted charged particles as accurately as possible, the magnet system bends their trajectories (i.e. due to the Lorentz force).

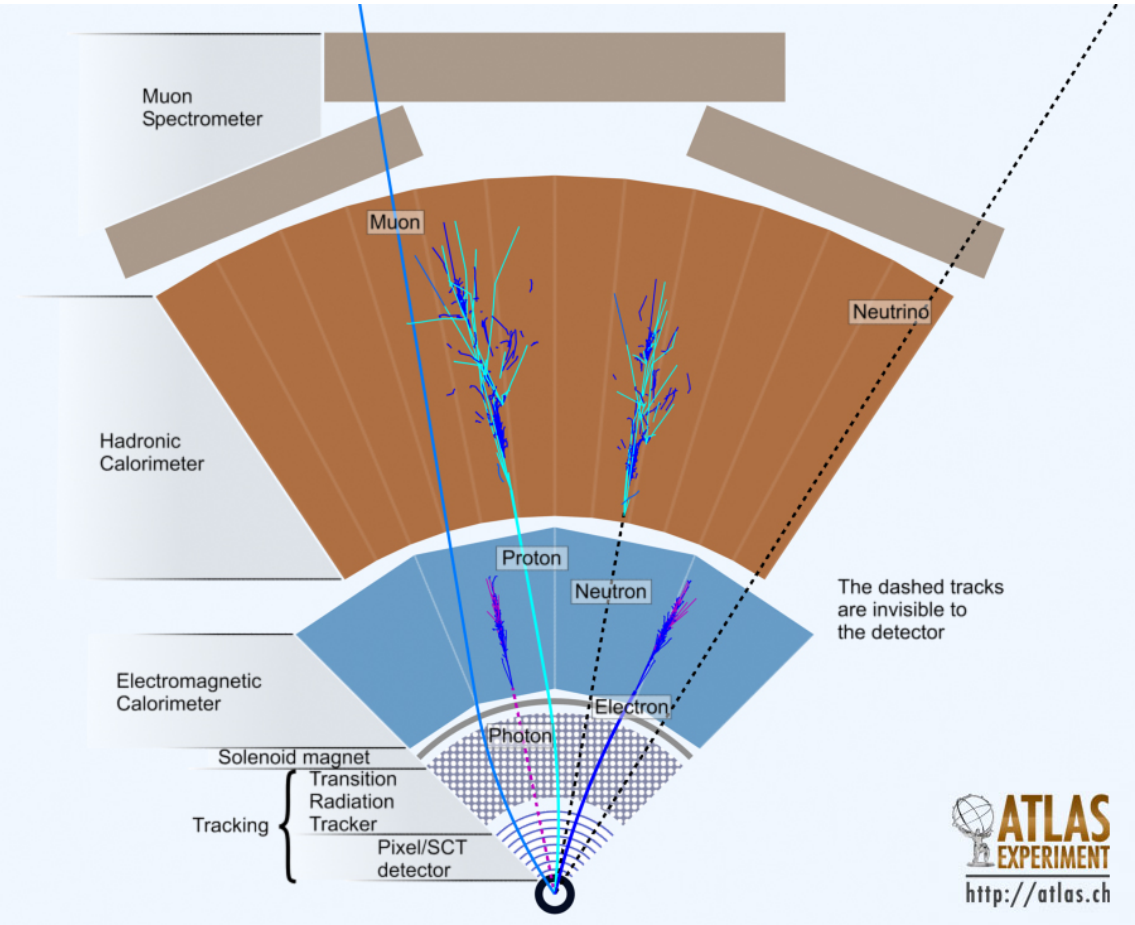


Figure 1.5: The behavior of different types of particles within the cross-section of the ATLAS detector [166].

1.4.1 The Inner Detector

The ATLAS IDET [17] [36] tracks the charged particles within the $|\eta| < 2.5$ region. It combines semiconductor detectors (*pixels* and *strips*) with gas tube detectors. It provides useful information for particle identification based on the starting points of the trajectories. It is placed within the 2 T magnetic field produced by the Solenoid Magnet which curves the trajectories of the charged particles. Thus, the charge can be deduced from the curvature direction and the momentum from the amount of the curvature for each case. Its internal structure is detailed in Figure 1.6.

The Pixel Detector [82] is the IDET's component closest to the beam axis and the interaction point (i.e. $|\eta| < 2$). It has three concentric cylindrical layers (barrel type at radii 50.5, 88.5 and 122.5 mm) and three concentric disk layers at the end-cap regions (at 495, 589 and 650 mm distances). In 2014, the Insertable B-Layer was installed between the Pixel Detector and the beam-pipe, at a radius of 33.5 mm [86]. The Pixel Detector is the ATLAS's component with the highest granularity (i.e. a pixel has a size of 50×400 μm). It contains 8×10^7 readout channels which amount to approx. 50% of the total channel number within the entire ATLAS detector. It is also subjected to the highest amount of radiation because of the proximity to the interaction point. Thus all its components are radiation hardened.

The Pixel Detector is surrounded by the Semiconductor Tracker [84] which contains nine end-cap disks on each end (between 854 and 2720 mm on the z axis) and

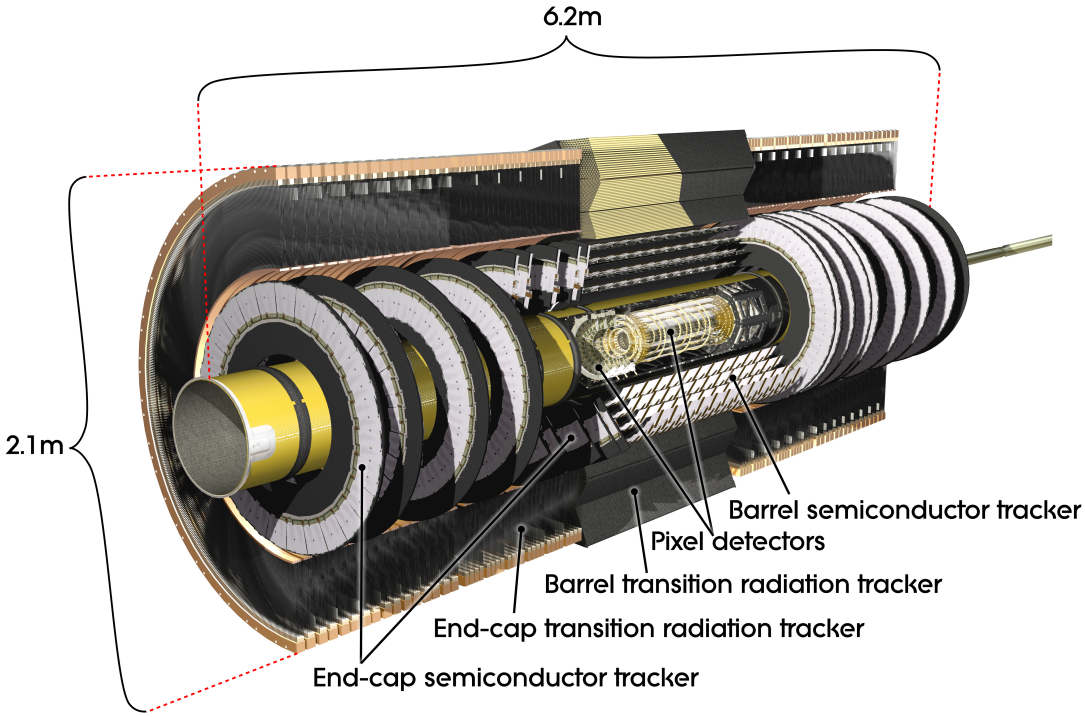


Figure 1.6: The internal structure of the ATLAS Inner Detector [163].

four barrels (at radii between 299 and 560 mm). Instead of pixels, it contains narrow strips (80 μm $\times$ 120,000 μm) amounting to 6.3×10^6 readout channels.

The outermost component of the ATLAS IDET is the Transition Radiation Tracker [195]. It contains tubes of 4 mm diameter and up to 150 cm in length filled with xenon gas, each having a fine filament in the center that has a -1500 V potential difference relative to the exterior. The passing of a charged particle causes the ionization of the gas and thus a small current pulse in the filament. There are approx. 300,000 such tubes interleaved. The paths of particles are determined from the hit patterns.

1.4.2 The ATLAS calorimeter system

The Solenoid Magnet, part of the magnetic system, encircles the ATLAS IDET. The ATLAS calorimeter system [15] is the next layer, having the purpose of absorbing and determining the energy of both neutral and charged particles. The energy is absorbed within a high-density material and results in cascades (called showers) of lower energy secondary particles which can further produce other particles and so on. The layers with absorbing material are alternated with layers containing active material placed in electric fields. The particle showers ionize the active material producing pulses of electric current which are proportional to the energy of the interacting particles. An example of such a process is the photoelectric effect. The current pulses are picked up by the electrodes of the readout channels which sample and quantize the signals. From inside to outside, there are two layers of calorimeters: electromagnetic and hadronic. The calorimeter system covers the area $|\eta| < 4.9$, has a total of approx. 188,000 read-out channels and is depicted in Figure 1.7 with its components labeled.

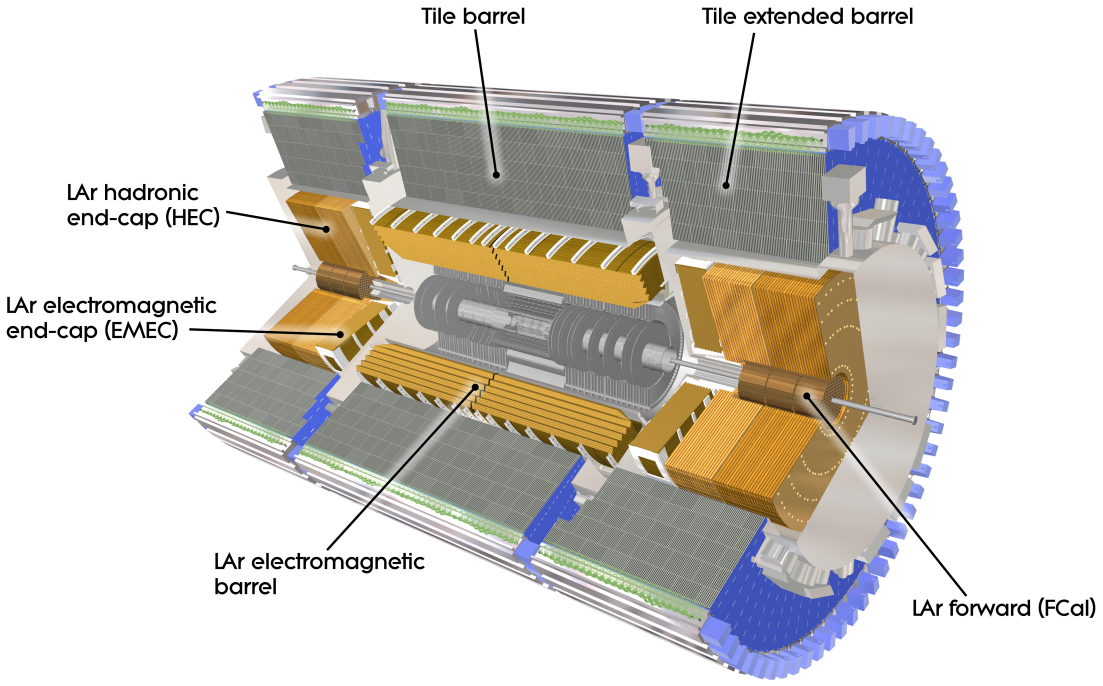


Figure 1.7: The internal structure of the ATLAS calorimeter system [162]. In the center, the Solenoid Magnet and the ATLAS IDET are located.

The electromagnetic calorimeter [39] [16] is situated in the barrel region of $|\eta| < 1.475$ and the end-cap regions corresponding to $1.35 < |\eta| < 3.2$. It is used to measure the showers produced by the incoming photons and electrons (see the depiction with Figure 1.5). The absorbing materials are lead and stainless steel while the active material is liquid argon (LAr) at 88.15 K (-185°C). It has 173,000 read-out channels.

The outer layer of the ATLAS calorimeter system is the hadronic calorimeter. It is divided into three sections: one central calorimeter with scintillator tiles (called Tile) and two lateral calorimeters (i.e. the Forward Calorimeter - FCal and the Hadronic End-cap Calorimeter - HEC) containing LAr.

The Tile [111] measures the energy of hadrons and is also used for the determination of the muons trajectories. A scintillator is a material having the property of producing luminescence (light) when it absorbs the energy of an incoming particle. Within Tile, the absorbing material is steel. It is represented by a barrel section and two extended barrels within the $|\eta| < 1.7$ region, totaling 9852 channels. The FCal [48] covers the region $3.1 < |\eta| < 4.9$ at a distance of 4.7 m from the interaction point. It measures the electromagnetic interactions with a copper module while the strong interactions (of the hadrons) are sensed with two wolfram modules. The HEC [117] is situated in the $1.5 < |\eta| < 3.2$ region around the FCal (see Figure 1.7) as two independent segments. It contains approx. 5600 read-out channels that measure hadrons with small polar angles (5 to 25 °).

1.4.3 The Muon Spectrometer

Muons are elementary particles with the same charge as electrons but a significantly higher mass (200 times), making them lose only a small part of their energy and only through ionization when passing through a material. Thus they are highly penetrating particles. The muons do not produce showers but do interact electromagnetically with other particles. As a result, the ATLAS muon system (i.e. the MS) [19] is placed in the outer parts of the detector in the regions of pseudo-rapidity $|\eta| < 2.7$, in a magnetic field of approximately 2 T produced by superconducting Toroid Magnets. The MS, depicted in Figure 1.8, ensures high accuracy tracking and energy measurement for the muons produced by the collisions of protons in the interaction point.

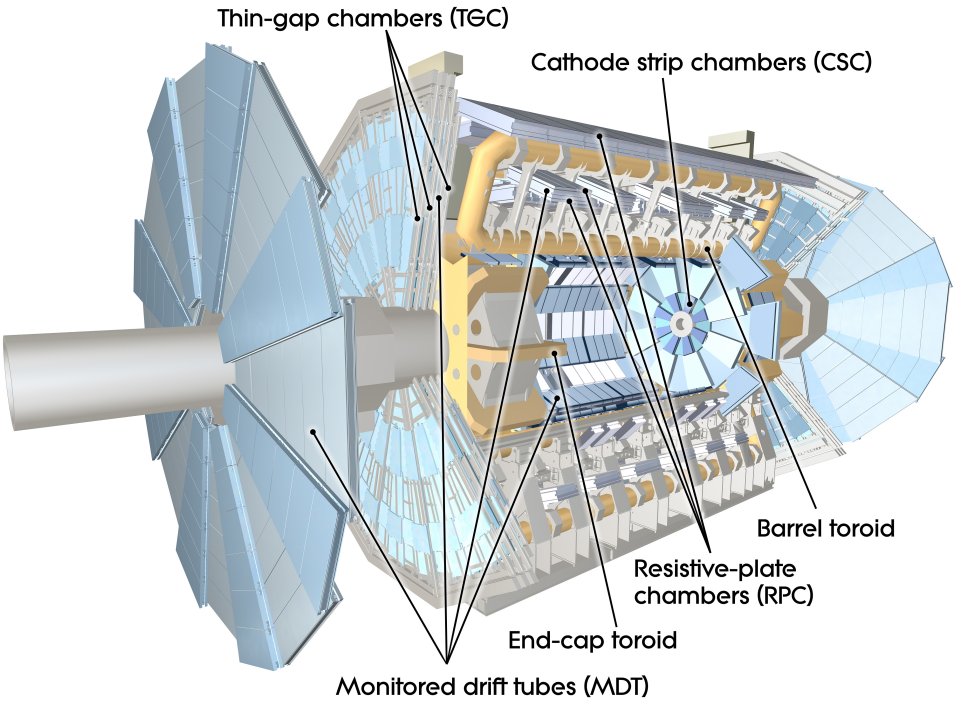


Figure 1.8: The internal structure of the ATLAS MS [164].

The muon detectors are chambers filled with gas mixtures kept in high-voltage electrical fields. They sense the muons passing though as electrical signals captured by electrodes and resulted from the gas ionization. The muon detectors are divided into two categories: fast trigger chambers that provide coarse information in several tens of nanoseconds and precision-trackers that provide the most precise measurements but with higher latency. The trigger detectors are used for the selection of the Region Of Interest (ROI) from the precision-trackers.

The MS has the following four types of detectors:

1. The MDTs [57] [200] are precision-tracking aluminum tubes placed in multiple layers in the region $|\eta| < 2.7$ (the region is $|\eta| < 2$ in the innermost end-cap layer). The tubes are filled with a mixture of carbon dioxide (i.e. CO_2) and argon. Between the aluminum wall and a central conductive wire, a 3000 V difference in electrical potential is present. In the entire ATLAS detector,

350,000 MDTs are installed in 1,150 chambers, providing a muon tracking resolution of 40 μm over a total area of 5500 m^2.

2. The CSCs [200] are precision-trackers that cover the region $2 < |\eta| < 2.7$, under the innermost MDT end-cap and help distinguish between ambiguous tracks. They consist of 32 chambers with four layers, filled with a carbon dioxide and argon mixture and use a 1900 V difference of potential. This results in 31,000 read-out channels and a resolution of 40 μm.

3. The RPCs [200] [81] are trigger chambers that cover barrel regions with $|\eta| < 1.05$. They consist of two conductive parallel plates distanced at 2 mm, with the space between them filled with a combination of gases. The voltage is 9600 V and there are 360,000 such channels with a resolution of 20 mm.

4. The TGCs MS are trigger chambers that cover the end-caps, in the regions $1.05 < |\eta| < 2.4$. The voltage is 2800 V and there are 320,000 readout channels with up to 2 mm resolution.

A quarter of the ATLAS detector section in the z-y plane is detailed in Figure 1.9, with emphasis on the MS's sub-systems. The RPCs and the TGCs are used for triggering the MDTs and CSCs precision-trackers. The innermost end-cap region of the MS formed by MDTs, CSCs and TGCs (i.e. between 7 and 8 m on the z axis and 0.7 and 4 m on the y axis) is called Small Wheel while the middle end-cap muon region (i.e. between 13 and 14.5 m on the z axis and 1.7 and 11 m on the y axis) is referred to as the Big Wheel.

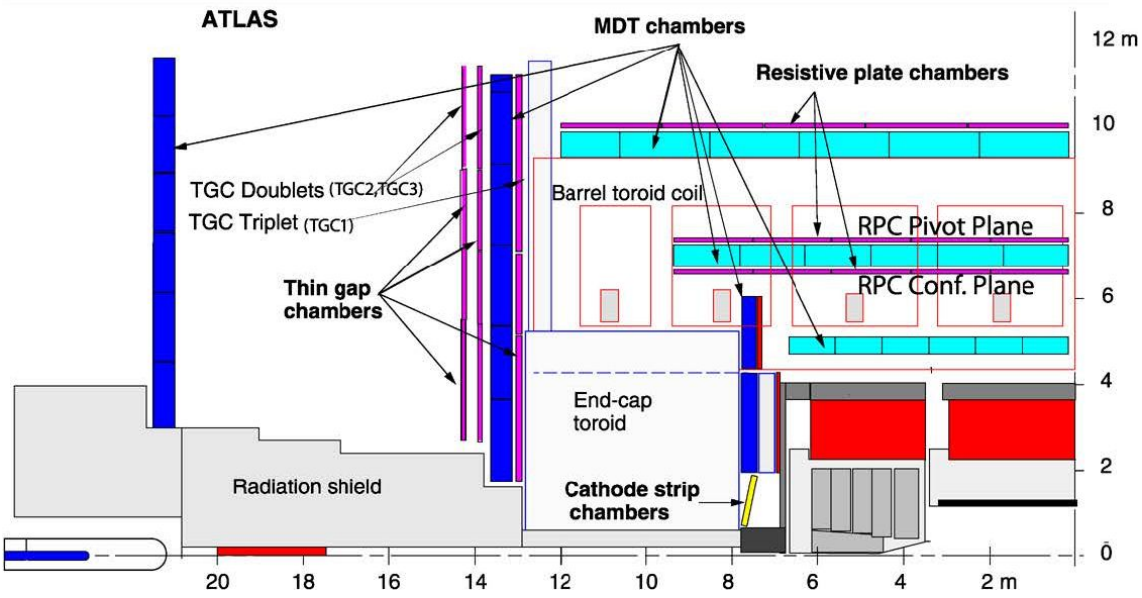


Figure 1.9: A quarter of the ATLAS detector section in the z-y plane, with the MS's sub-systems emphasized [83]. The horizontal axis is z, while the vertical one is y.

1.5 The ATLAS TDAQ system

Considering the LHC's 40 MHz BC rate and the large number of readout channels within the ATLAS detector (approx. 100 million) an enormous amount of data is produced (in the order of PB/s [107]; 1 PB = $2^{50} = 1,125,899,906,842,624$ bytes). Since it is impossible to transmit, process and store this entire amount, the ATLAS TDAQ system is responsible for interpreting the detector signals and determining in real-time the ROIs for each BC. Event filters aim to reduce the data rate further since only a few of the collisions produce physical events of interest. Based on these decisions, the TDAQ system converts the selected detector signals into a dataset that is transmitted, stored and later analyzed. The resulted data rate is approx. 1 GB/s [74].

A block diagram of the ATLAS TDAQ system as used in Run 2 (i.e. between 2015 and 2018) is depicted in Figure 1.10 with emphasis on triggers and dataflows. The first filtering element is the hardware-based first level trigger - Level-1 (L1) while the second filtering element is the software-based High-Level Trigger (HLT). The L1 Accept commands are generated by the Central Trigger Processor (CTP) with a maximum rate of 100 kHz and a 2.5 μs latency, based on the L1 calorimeter (L1 Calo), the L1 Muon information and the L1 Topological (L1 Topo) trigger. The analog signals from the calorimeter detectors are digitized by the Preprocessor. The result is transmitted in parallel to the Cluster Processor (CP) and the Jet Energy sum Processor (JEP). The CP searches for electrons, photons and τ-leptons with characteristics above a configurable threshold. The JEP computes sums of total and missing energies and searches for jets (i.e. narrow cone-shaped ensemble of hadrons and other particles resulted from scattering processes).

For the end-cap regions, the L1 Muon trigger matches the hits from the inner and outer TGCs layers between them and with the tile calorimeter data. For the barrel sections, the RPCs hits are used. The L1 Muon trigger information is sent to the CTP through the L1 Muon Central Trigger Processor Interface (MUCTPI). The L1 Topo trigger forms geometric and kinematic combinations of L1 Calo and L1 Muon trigger objects and applies topological requirements.

To limit the rate of L1 Accept commands, the CTP applies a limiting mechanism called *dead time*. This means that the allowed number of L1 accepts in a configurable number of consecutive BCs is limited (called *dead time*) to avoid the overflow of the buffers in the front-end read-out part. Also, overlapping read-out windows are avoided by limiting the minimum time between two consecutive L1 Accept commands (mechanism called *simple dead time*).

The on-detector front-end electronics form events based upon the L1 selections using all detectors. The events are then transmitted to the Read-Out Drivers (RODs) which apply initial processing and formatting. The results are then passed to the Read-Out Systems (ROSs) which act as buffers until the HLT requests data.

During the formation of the L1 trigger, ROIs are also defined and passed to the HLT. Based on this information, the HLT runs reconstruction algorithms on dedicated computing farms. These algorithms are similar to what is run when the stored data is analyzed but they are much simplified to decrease the execution times. Based on the results, the output data rate is further reduced. The events accepted by the HLT are transmitted to the Tier-0 facility (the CERN Data Center) for storage and analysis. During Run 2, the average HLT rate was 1.2 kHz corresponding to a

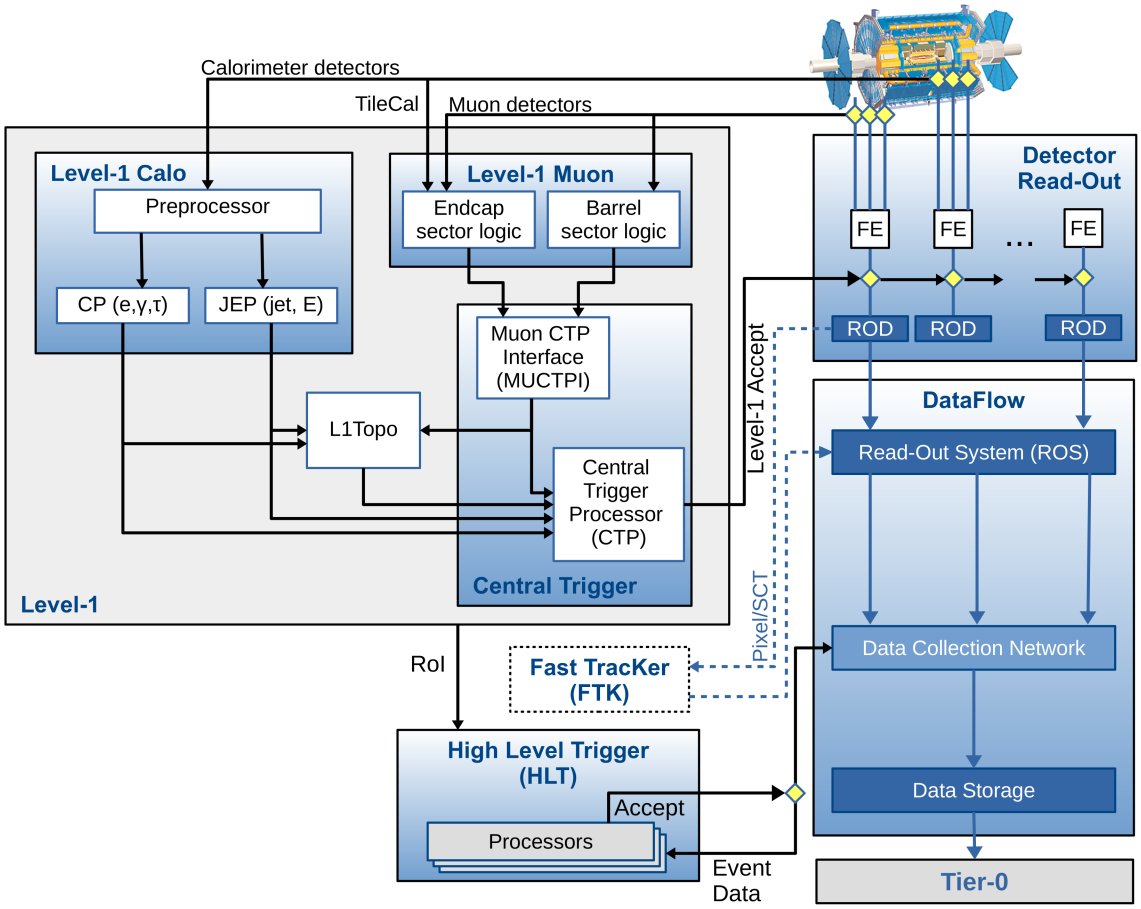


Figure 1.10: The ATLAS TDAQ system during Run 2 (2015 - 2018) showing the components contributing to the triggers and dataflows [85].

1.2 GB/s data throughput to the storage. The HLT latency depends on the number of pile-up interactions (i.e. the multitude of interactions happening in the same BC), being on average in the order of a few hundred ms.

The hardware-based Fast TracKer (FTK) determines the tracks within the IDET for the HLT. It was not used during Run 2 since it was being commissioned.

1.6 The upgrade of the ATLAS Experiment

The first LHC tests were successfully conducted in September 2008 [6]. The first operation run (called Run 1) began in November 2009 and ended at the beginning of 2013. The integrated luminosity of the proton-proton collisions during Run 1 was ≈ 30 fb^{-1} with the protons being accelerated up to 4 TeV [40]. The collected data led to the confirmation of the Higgs boson [99] [100]. Then, the LHC was shut down for a 2-year long upgrade and maintenance process called Long Shutdown 1 (LS1). The upgrade prepared LHC for its second operational run, called Run 2, which aimed at reaching a 14 TeV proton-proton collision energy (i.e. the bunches being accelerated at 7 TeV) [64]. Run 2 lasted from April 2015 until December 2018, a maximum energy of 13 TeV for the proton-proton collisions was reached and an integrated luminosity of ≈ 160 fb^{-1} within ATLAS and CMS [198] was accumulated.

The current update and maintenance process, called the Long Shutdown 2 (LS2)

[186], started at the beginning of 2019 and aims at implementing the first step (Phase-I) of the High Luminosity LHC (HL-LHC) project [44]. The main objective of HL-LHC is the reaching of at least 3000 fb^{-1} integrated luminosity over 10 years of operation (i.e. a 10 times increase). The modifications of the ATLAS detector related to HL-LHC are presented in [191]. LS2 is scheduled to end in 2022 and it is followed by Run 3. The Long Shutdown 3 (LS3) will implement the second and last upgrade part (Phase-II) for HL-LHC. HL-LHC is planned to begin operation within the second half of 2027.

The luminosity increase means that the detectors and the associated on-detector electronics will be subjected to increased nuclear radiation. At the same time, they must provide increased performance for the higher collision rates. Thus, the detector technology, the TDAQ systems and the controlling and monitoring software tools must be optimized and adapted. The most changes will be implemented for the ATLAS IDET, the end-caps of the calorimeters and the ATLAS MS.

In Phase-I of the HL-LHC upgrade, LINAC4 replaces LINAC2 and the energy of the PSB is increased. The LHC's nominal design instantaneous luminosity of $L = 10^{34}\ \mathrm{cm}^{-1} \cdot \mathrm{s}^{-1}$ will double or even triple. The FTK will be used for improving the trigger system and the *Small Wheel* end-cap regions of the MS will be replaced with the *New Small Wheel (NSW)* [135] components. The former aims at reducing the processing latency for the tracks. The latter is motivated by the large rate of false triggers produced by the Small Wheel (approx. 90% in the 2012 data [135]) and its performance degradation (i.e. resolution and efficiency) at high energies and high instantaneous luminosity.

The coverage area of the NSW is described by $1.3 < |\eta| < 2.7$ while the remaining area of the inner muon end-cap, $1.0 < |\eta| < 1.3$, maintains its existing detectors. The NSW consists of new trigger and precision tracking detectors that have increased spatial and time resolutions and can function at high rates. The two new technologies are the small-strip Thin Gap Chamber (sTGC) [98] and Micro-Megas (MM) [129] and are both used for triggering and tracking. The muon tracks originating in the interaction point can thus be determined more precisely. As a result, the rate of false triggers will be reduced. The details of the entire ATLAS MS upgrade are presented in [29].

In Phase-II of the HL-LHC, the entire ATLAS IDET will be replaced by the new *Inner Tracker (ITk)* system [110] [28] which consists of all-silicon detectors with increased granularity and radiation hardness. A new silicon-based detector system will be added in the $2.4 < |\eta| < 4.0$ region: the High-Granularity Timing Detector [34]. The ATLAS calorimeter system also has planned changes [178] [31] [27]. Within the ATLAS TDAQ, the current L1 trigger will become Level-0 (L0) and a potentially new L1 will implement a more complex selection. The rate and latency for both these levels will increase and thus the readout systems will also be upgraded to offer larger buffering spaces and increased throughput. The details of the Phase-II ATLAS TDAQ system upgrade are presented in [30] considering a single level trigger system. The requirements of both the single and two-level ATLAS TDAQ system are elaborated in [177]. During the elaboration of this thesis, the community decided that a single level trigger will be employed for Phase-II ATLAS [132]. All the work presented in this thesis was performed considering both modes of operation. However emphasis is placed on the two-level trigger mode of operation since it implies higher complexity and increased trigger rates and latencies.

1.7 The NSW TDAQ Context

The NSW contains in total approx. 2.45×10^6 detectors of MM (approx. 2,100,000 [135]) and sTGC (approx. 332,000 [135]) types. While both types contribute to both the trigger formation and the precision tracking, the MM detectors are used mainly for the track reconstruction due to their high spatial resolution (up to 100 μm) [185] and the sTGC for determining the trigger candidates due to their capability of identifying a singular BC [135].

The block diagram of the associated NSW electronics is depicted in Figure 1.11 and all the acronyms are explained in Table 1.2. The VMM is not an acronym, but the standalone name for the centerpiece detector read-out ASIC.

Since the LHC beams in Run 3 and beyond will consist of bunches that collide every 25 ns, the ATLAS TDAQ system will be paced by a synchronous 40 MHz clock signal called the BC or LHC clock. For serial data transmission and operations that require a faster rate, clock signals with higher frequencies, multiples of 40 MHz, are generated (e.g. the NSW TDAQ system's Read-Out (RO) clock signals are 160 MHz). There are 3564 particle bunches in one LHC orbit of which up to 2808 contain protons, the others being empty due to the required rise times and beam dumps in different LHC systems [177]. Thus, a 12-bit BCID (BC IDentification) value is used to uniquely annotate them [177]. The bunches complete an LHC revolution in 88.924 μs (i.e. an orbit frequency of 11.2495 kHz) [177]. The orbits are counted with a 32-bit OrbitID (Orbit IDentity) value which repeats with a period of approx. 100 hours. Since the LHC stable beam runs (i.e. the operational phases in which conditions are stable, particle collisions are happening and the detectors are recording data) last approx. 10 - 15 hours [177], the orbits are uniquely identified by the OrbitID value. The triggered events are labeled with the BCID and OrbitID pair, a 32-bit Level-0 IDentifier (L0ID) and/or a 32-bit Level-1 IDentifier (L1ID) counter value (if the second level of trigger is used in Phase-II). A subset of either the OrbitID, L0ID and/or the L1ID (i.e. consisting of Least Significant Bits - LSBs) can be transmitted to or from the front-end ATLAS TDAQ systems to conserve bandwidth as long as the full values can be inferred.

The ATLAS CTP determines the ROIs within the BCs of interest, based upon the detector data fed through detector-specific electronics, processed by dedicated trigger processors and transmitted on dedicated links. Through the Time Trigger and Control (TTC) system the LHC clock is distributed; the BCIDs, OrbitIDs, L0IDs and L1IDs values from all ATLAS systems and subsystems are synchronized and the BCs of interest are selected within the ROIs. The read-out system is responsible for managing the detector data fed by the detector-specific electronics and based upon the received trigger decisions, constructing, buffering and transmitting event fragments tagged with the above-mentioned identifiers. The time interval between the selected BC and the arrival of the corresponding selecting trigger command is called trigger latency. According to [177] which gives high-level requirements for the ATLAS TDAQ system for the Phase-II HL-LHC upgrade, the hardware ATLAS trigger will consist of either one (L0) or two (L0 and L1) levels. When operating alone, the L0 and will have a 10 μs latency and a 1 MHz maximum rate. In conjunction with L1, the maximum L0 rate will be 4 MHz. The L1 has a maximum rate of 800 kHz and 35 μs maximum latency. The NSW TDAQ system is compatible with both modes of operation. More details are presented in Chapter 2.

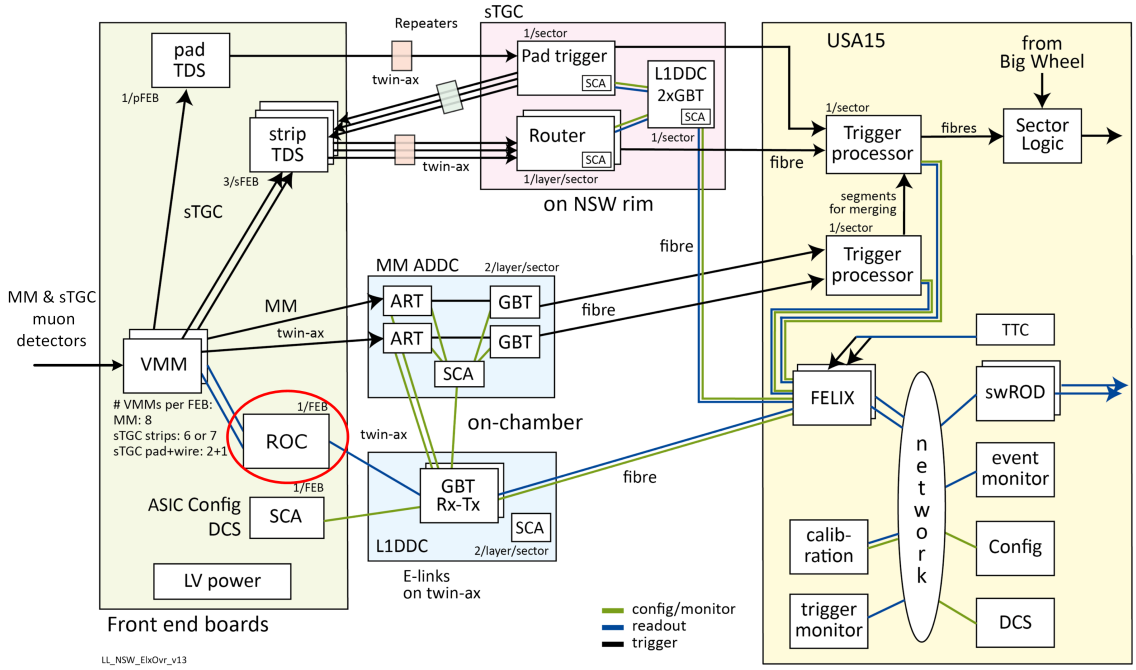


Figure 1.11: The overview of the NSW TDAQ system [141] - edited. One VMM ASIC digitizes the signals from up to 64 MM or sTGC muon detectors. All acronyms are explained in Table 1.2.

Acronym	Description
ADDC	ART Data Driver Card (ADDC) [216] [217] - on-detector printed circuit board used to transmit trigger data from the MM detectors to the MM Trigger Processor; contains two ART chips, two GBTx ASICs and one SCA chip.
ART	Address in Real-Time [1] - ASIC used to process the ART signals from 32 VMM3 ASICs (four MMFE8s). The ART signal represents the 6-bit address of the first VMM channel that has a voltage pulse above an adjustable threshold. The ART chip provides the complete addresses (i.e. 5-bit VMM address, 6-bit VMM channel) of up to 8 detector channels that experienced hits, alongside the corresponding 12-bit BCID with a latency of up to 500 ns [216] [217].
FEB	Front-End Board - on-detector PCB containing the read-out electronics for the NSW detectors. Depending on the type of detectors being served there are two types of FEBs. One MM FEB (MMFE8) contains eight VMMs, one ROC and one SCA ASIC. For the sTGC detectors, there are two variants of the FEB: sFEB (for the sTGC strips) and pFEB (for the sTGC pads and wires). One pFEB contains 3 VMMs, one ROC, one TDS in pad mode and one SCA. One sFEB contains six, seven or eight VMMs, one ROC, one TDS in strip mode and one SCA.
FELIX	Front End LInk eXchange - Off-detector FPGA-based (Field Programmable Gate Array) network interface for the ATLAS front-end electronics [49].
DCS	Detector Control System - configuration, calibration and monitoring of the detectors and the associated boards and electronics.
GBT or GBTx	GigaBit Transciever - radiation tolerant ASIC used to multiplex/demultiplex data from/to multiple sources/destinations into/from a full-duplex high-speed (3.2 - 4.48 Gb/s effective bandwidth) multipurpose optical link [153] [140].

L1DDC	Level-1 Data Driver Card [119] [150] - on-detector PCB that aggregates the read-out and monitoring data from multiple FEBs and transmits it to FELIX using GBTx ASICs. It also distributes the TTC stream and configuration data. On L1DDC board contains two GBTx ASICs and two VTRxs.
LV	Low-Voltage - supplied by Low-DropOut (LDO) voltage regulators.
ROC	Read-Out Controller [79] - on-detector radiation-tolerant ASIC - the subject of this thesis.
SCA	Slow Control Adapter [114] - radiation-tolerant ASIC used to distribute/acquire control/monitoring signals from the on-detector electronics and the detectors.
TDS	Trigger Data Serializer [196] - on-detector, radiation-tolerant ASIC that performs sTGC pad-strip trigger matching and serialization and transmission of the resulting data to the signal Router board.
TTC	Time Trigger and Control [53] - a distribution system from the ATLAS CTP to the detector read-out electronics for adjusting their timing and providing trigger and control commands.
USA15	Underground Service ATLAS - the first of the four big LHC underground halls. It is situated 90 m from the ground level and acts as the ATLAS service hall.
VMM	Standalone name, not an acronym for the centerpiece detector radiation-tolerant read-out ASIC that interfaces with both types of NSW detectors.
VTRx	Versatile Transceiver - CERN custom-made optical transceiver [47].

Table 1.2: The descriptions of all the acronyms from Figure 1.11.

The MM and sTGC readout path consists of the following components: VMM [94], Read-Out Controller (ROC) [79], Level-1 Data Driver Card (L1DDC) [119] [150] GigaBit Transciever (GBTx) [153] [140] and Front End LInk eXchange (FELIX) [49]. The MM trigger scheme consists of: VMM, Address in Real-Time (ART) [1], ART Data Driver Card (ADDC) [216] [217] GBTx, MM Trigger Processor, FELIX and sTGC Trigger Processor. The sTGC trigger path consists of VMM, pad Trigger Data Serializer (TDS) [196], Pad Trigger Extractor board (i.e. Pad trigger in Figure 1.11), strip TDS, Router board, sTGC Trigger Processor and FELIX.

The main radiation-tolerant ASIC for reading and interpreting the signals from both types of detectors is the *VMM*, which represents an Amplifier Shaper Discriminator (ASD). An *amplifier* conditions the relatively small analog signals from the detectors improving the Signal-to-Noise Ratio (SNR) and adapting the input and output impedances [184] [112] [108]. The *shaper* tailors the analog signal for an Analog-to-Digital Conversion (ADC) (e.g. adapts the pulse duration to the conversion time, rounds peaks, integrates the input signal, etc.) [184]. The *discriminator* is an electronic circuit that compares the analog input signal with a configurable threshold producing a digital output, acting as a comparator or 1-bit ADC [112] [184]. The third version of the VMM ASIC (i.e. VMM3 [130]) will be used from Run 3 forward. One VMM3 chip contains 64 distinct detector read-out channels providing peak amplitude and time measurements via configurable charge amplification, discrimination and precise Analog-to-Digital Conversion (ADC). The voltage peak amplitude resolution is 10-bit and its time of occurrence relative to the BC clock is represented as an 8-bit word. The VMM3's read-out digital logic buffers and aggregates the data from its channels based upon the received L0 trigger selection commands and transmits the resulting L0 events, tagged by the corresponding OrbitID - BCID pair, to the ROC ASIC. One ROC chip collects and buffers the L0

events from up to eight VMM3 ASICs (i.e. data from up to 512 NSW detectors), implements the L1 selection (ultimately not used [132]), constructs more complex packets with aggregated data, decodes the TTC stream and distributes its commands and supplies clock signals. It reduces the number of necessary data links on the read-out path. As it is highly configurable, it allows bandwidth utilization optimization. **All the contributions of this thesis are related to the ROC ASIC, therefore this chip is extensively described in this thesis. At least 4875 ROC chips will be installed and function concurrently within the NSW TDAQ system.**

For the MM technology, groups of eight VMM3 ASICs, one ROC and one SCA chip are placed and interfaced together on PCBs called MMFE8s (MM FEB 8 VMM3s). For the sTGC chambers, depending on the parts of the detectors being read, i.e. the pads or the strips, two types of Front-End Board (FEB)s are used: pad FEB (pFEB) and strip FEB (sFEB). A pFEB contains three VMM3s, one ROC, one TDS in pad mode and one SCA ASIC while an sFEB contains six, seven or eight VMM3s, one ROC, one TDS in strip mode and one SCA ASIC. All three FEB types are interfaced on the read-out path (i.e. the data output links of the ROC) with the L1DDC boards through E-links [62]. One L1DDC board employs 2 GBTx chips for data aggregation from up to 8 FEBs. The ROC, TDS and VMM chips are configured and monitored through the SCA chip. Since the high-speed optical connection of the GBTx ASIC is bidirectional, the TTC and configuration data is distributed through the L1DDC to the associated FEBs. The L1DDCs are interfaced with FELIX, an FPGA-based data router that transmits the read-out data to the RODs. FELIX also distributes the TTC and DCS signals.

On the trigger path, the VMM3 can provide faster but coarser measurements of the detector signals. Each of its channels contains a fast comparator circuit with a configurable threshold. The single-bit output signal from each comparator is fed directly to a VMM3 output pin. Thus, a VMM3 chip provides 64 Time-over-Threshold (ToT) flags. Each VMM3 channel also contains a faster 6-bit ADC circuit for the amplitude measurement.

For the MM technology, if the detector signal crosses the selected comparator level, the VMM3 sends the channel address of the first hit for that BC to the ART chip on a dedicated interface. An ADDC board contains two ART chips and two GBTx ASICs for interfacing to the NSW MM Trigger Processor (TPROC). The ART chip forms and transmits packets containing the first hit channel address and its timestamp from up to 8 VMM3 ASICs. Its real-time property results from its 500 ns maximum processing time.

For the sTGC detectors, the ToT and the 6-bit ADC data are used (the ToT for pFEB and the 6-bit ADC for sFEB). The pad TDS monitors the ToT signals, tags them with the corresponding BCID and transmits the result to the Pad Trigger board. This board contains an FPGA-based processor that estimates muon ROIs based upon the ToT data from multiple pad TDS chips. The strip TDS manages the 6-bit ADC data from the associated VMM3 chips. The Pad Trigger processor signals the determined ROIs to the strip TDS chips which, as a result, then send the corresponding ADC data to the NSW sTGC TP through the Router Board.

Both the MM and sTGC TPs are FPGA-based, being interfaced with each other. They collect the trigger primitives (strip charges or strip hits), calculate charge centroids, find and fit track segments and remove duplicates. The resulting candidates

are sent to the FPGA-based Sector Logic board that finds matches with the Muon Big Wheel hits and then transmits the muon candidates to the MUCTPI.

1.8 Objectives

This Ph.D. thesis represents the culmination of the author's work within the CERN-ATLAS-NSW context.

The objectives of this thesis were the following:

1. design of the ROC's logic and its quality assurance, following all its requirements as accurately as possible, without introducing bottlenecks (i.e. the performance must be limited only by the used clock signals, interfaces and associated protocols) and considering that the resulting ASIC must be operational from the first manufacturing (i.e. a single tape-out without bugs or misunderstood requirements that may require re-designs that attempt to correct or mitigate them);
2. correct and in-time implementation of the design resulted at objective 1 into the targeted Complementary Metal Oxide Semiconductor (CMOS) technology, with all the available checks, analyzes and verification steps being performed and indicating no flaws;
3. functional validation of the manufactured design;
4. performance measurements for the manufactured design;
5. quality control of the mass-fabricated chips;
6. real-world assessment of the ROC's implemented techniques for mitigation of temporary radiation-induced faults (i.e. ROC design radiation qualification);
7. integration support for the ROC ASIC;
8. identification and pursuit of new research opportunities as a result of the activities performed for achieving the above-mentioned objectives;
9. dissemination of the obtained results into journal and conference papers, this thesis and other documents (e.g. presentations and documents presented within the CERN-ATLAS-NSW context);
10. participation and completion of relevant courses provided by the Doctoral School and other institutes, attending CERN's seminars, summer and winter schools, workshops, self-study and self-documentation.

1.9 Thesis Outline

In Chapter 2 the ROC's context, main requirements, interfaces, top-view and detailed architectures are presented. Implementation details are included and a queueing theory model is determined. The steady-state condition as a function of the input data characteristics and the used configuration is deduced. The main algorithms

that construct and transmit the output data are presented using pseudo-code and their hardware implementation is depicted using state diagrams, state-transition tables and schematics. The chapter covers the scientific objectives 1 and 2 from the previous section.

Chapter 3 starts by describing what the distinct notions of ASIC verification and testing are. An exhaustive description of the developed ROC's digital functional test setup follows. This custom setup is hardware-based and was used for the ROC's design validation and mass-testing. The obtained results are presented and analyzed. The steady-state condition from Chapter 2 is validated in the real world through testing. An improved method to achieve the relatively complex synchronization of the interfaces within the asynchronous but sequential test setup system is detailed and empirically validated. Thus, the chapter fully covers the scientific objectives 3, 4 and 5 and partially objective 8 described in the previous section.

Chapter 4 starts with a summary of the types of radiation-induced faults within digital circuits and the basic techniques for mitigating them. The implemented mechanisms within the ROC design for mitigating the temporary change of state of a sequential element as a result of an incident ionizing particle are detailed next. The ASIC's functionality within an environment with controlled neutron irradiation was tested. The used test setup, the characteristics of the incident neutron beam and the obtained result are extensively presented. Based upon the observations, estimations are made for Phase-II of HL-LHC. New theoretical scenarios of induced change of state are presented, simulated and explained. This chapter fully covers thesis objective 6 presented in the previous section.

In Chapter 5 the idea of a new implementation for an existing tool often used within digital designs to increase observability is presented. During the ROC's design validation and radiation qualification, some limitations of the existing implementation of this tool were observed. These limitations are described and the proposed solutions are presented. Then, the proposed technical implementation is presented and its advantages and disadvantages are emphasized. Preliminary experimental results are included. After that, the found solutions are compared with relevant patents. The tool implementation aligns with the thesis subjects since it represents a packet processor, it can be used within complex physics experiments for TDAQ and it implies digital design. This chapter falls within objective 7 of the previous section.

In Chapter 6, the last, the conclusions are drawn. A summary of all the activities related to the thesis scientific and training of human resources objectives, including objective 7 from the previous section, is presented. Next, the author's contributions to all the scientific matters are listed. The dissemination of the results follows.

Except for the first and last, all the chapters end with a section that concludes the subjects presented within them.

Chapter 2

The Read-Out Controller (ROC)

This chapter is dedicated to the ROC's requirements, context, architecture, implementation, model and algorithm. It starts with a section describing the main specifications, extracted from the NSW Electronics group's VMM3-ROC joint requirements document [123]. The chapter continues with a section about the ROC context and its top-view architecture. The next section describes the ROC architecture in detail, with emphasis on its interfaces, data formats and implemented mechanisms for congestion and flow control. It includes a rationale about the chosen FIFO (First-In-First-Out) sizes and a short presentation of the resulted layout and the final packaging. After that, a queueing theory model for the ROC is presented. Based upon the described IO (Input-Output) interfaces and data formats, formulas for its maximum performance are deduced. This model is validated in the next chapter which extensively presents the ROC testing and validation. The current chapter continues with a presentation of the main algorithm for constructing L1 events in response to the received L1 triggers. The triggers dedicated to the ROC are called L1 triggers, the resulting output packets are L1 events while the input VMM3 packets are L0 events, regardless of the use of one or two stages of hardware triggers within the ATLAS TDAQ. The main algorithm is presented in pseudocode format in Appendix A.1 and its hardware implementation in the form of a Finite State Machine (FSM) is detailed in Appendix A.2 as a state diagram, transition tables and the associated logic schematic. The chapter ends with a section that describes the procedures that resulted in the verified ROC design submitted to fabrication, with emphasis on the ones related to its digital part. The research presented in this chapter was disseminated in [79] [173] and [174].

2.1 ROC main specifications

The ROC ASIC represents an on-detector custom real-time packet processor which is a key part of the NSW TDAQ system because it offers several advantages. The latency requirements for the L1 trigger can be relaxed due to the its relatively large buffers, compared to other context ASICs (e.g. VMM3). In conjunction with the implemented flow and congestion control mechanisms this minimizes data loss. Being a highly configurable concentrator, the ROC is used to aggregate data from multiple channels with different amounts of throughput. The bandwidth utilization is optimized since not all NSW detectors will produce the same quantity of data.

The main specifications for the ROC resulted from the NSW Electronics group's VMM3-ROC joint requirements document [123] are:

1. The ROC receives 8b10b [199] encoded data representing L0 packets from up to eight VMM3 ASICs on separate input channels called VMM Capture. The data from one VMM3 ASIC are DDR (Double Data Rate), serialized on two SLVS (Scalable Low-Voltage Signaling) [54] transmission lines using a RO clock with a frequency of 160 MHz. The bits are alternatively sent on the two lines (waveforms are presented in Section 2.3.1, Figure 2.10). The resulting total bandwidth is 640 Mbps (640×10^6 bps). The receiving VMM Capture channels should deserialize the incoming data, determine its alignment, decode it, determine its correctness and buffer the resulted L0 packets into SRAM (Static RAM) with a FIFO strategy. These input interfaces and the associated data format are presented in sections 2.3.1 and 2.3.2, respectively.

2. The ROC receives from the ATLAS CTP, through FELIX and the L1DDC GBTx ASIC, the TTC stream and the LHC (i.e. the 40 MHz BC) clock signal. Based upon the supplied BC clock signal, the ROC generates its internal BC and 160 MHz RO clock signals that pace its logic. One BC and one RO clock signal should be supplied to each of the eight associated VMM3 ASICs. In addition, the ROC should supply either eight RO clock signals for the data transfer between the VMM3 ASICs and the MM ADDC ART chips (called ART clock signals) or four BC clock signals to TDSs (on sTGC pFEB and sFEB) in conjunction with other control signals that are described below. All the ROC supplied clock signals are independent and phase-adjustable in steps of at least 1 ns. The TTC stream is organized into bytes whose bits represent commands that synchronize the BC and event counters and select the BCs of interest. The bytes are not 8b10b encoded but are directly serialized on a single SLVS DDR lane also using a 160 MHz RO clock signal (resulting in a 320 Mbps = 320×10^6 bps stream of bits). Thus, in each BC up to eight commands can be transmitted. An additional ninth command is formed by the same asserted bit in two consecutive BCs. This interface is further detailed in Section 2.3.1. A TTC byte is delimited within the incoming TTC stream by the positive edge of the input BC clock signal. It contains the following nine commands:

   (a) L0A - Level-0 Accept (i.e. the L0 trigger). It is supplied to all the associated VMM3 chips as pulses synchronous to the corresponding supplied BC clock signals or with a configurable phase shift relative to them. When the VMM3 reads a set L0 pulse it means that the current values of its internal 12-bit BCID and 2-bit (i.e. the LSBs) OrbitID counters represent a BCID of interest that is selected and processed by its L0 logic.

   (b) SR - Soft-Rest. Represents an active high reset signal that clears every Flip-Flop (FF) within ROC and its associated VMM3s except the configuration registers. Within ROC, the SR is synchronous to the internal BC clock signals. The output SRs are phase adjustable relative to the corresponding output BC clock signals.

   (c) ECR - Event Counter Reset. Used solely within the ROC in this context (i.e. not forwarded). The ROC logic implements a 16-bit (the LSBs)

L1 event counter that is incremented at each set Level-1 Accept (L1A) command. This counter is cleared upon the arrival of a set ECR bit.

(d) BCR - BCID Counter Reset. It is supplied to all the associated VMM3 ASICs and optionally to four TDS chips (if the case when the ROC does not supply the eight ART clock signals). The supplied BCRs are synchronous to the supplied BC clock signal or can have a configurable fixed phase shift relative to it. In all the receiving chips, when this bit is set a configurable value is loaded within the 12-bit BCID counter. When not cleared, the BCID counter increments in every BC clock cycle until it reaches the also configurable threshold. Thus all the BCID counters from all the chips are synchronized and when an L0A or L1A command arrives, the same values are selected in all of them regardless of the different (but constant) propagation delays of the TTC stream.

(e) OCR - OrbitID Counter Reset. It is formed by two consecutive set BCR commands. Thus, it is distributed to all the recipients of BCR without employing separate transmission lines. When the OCR condition happens, the BCID counter is loaded with the set value, just as for a single BCR. The difference is that the OrbitID counter (which normally increments its value every time the BCID counters reaches its rollover value) keeps its value until the BCID reaches the set rollover value and then its value is cleared.

(f) L1A - Level-1 Accept (i.e. the L1 trigger) is used solely within the ROC in this context (i.e. it is not transmitted further). When an L1A high pulse arrives the current values of the BCID, OrbitID and L1ID are pushed in the L1 trigger FIFOs of the four Sub-ROC (SROC) modules (described below). These values are then used to search for matching data within the associated VMM Capture channels. Section 2.5 details this procedure.

(g) SCA reset - asynchronous active-high reset signal for the SCA ASIC.

(h) EC0R - Event Counter L0 Reset. This command was reserved to clear the L0 event counters (similar to what ECR does for the L1ID) in both the ROC and its associated VMM3s but was eventually never used since the L0 events do not contain any data related to their corresponding L0IDs (not even a subset) and the ROC does not implement any logic to determine the correspondence between the L1A and the L0A commands.

(i) TP - Test Pulse used solely within the associated VMM3 ASICs. It triggers the generation of L0 data for debugging and testing purposes without the need for interfacing the chip with actual NSW detectors. The delay of the TP signal relative to the originating BC should be configurable in 3.125 ns (i.e. clock cycle duration of a 320 MHz clock signal) steps up to one BC (i.e. 25 ns).

3. The ROC should contain four SROC modules or channels that have predefined but distinct VMM Capture channels associated with them. Several predefined association configurations are provided in the VMM3-ROC joint requirements document however they are not presented here since a fully configurable crossbar module was designed to route the necessary signals between the eight VMM Capture modules and the four SROC channels (see Section 2.3). One

SROC can have more than one VMM Capture FIFOs associated with it, but a VMM Capture FIFO cannot be linked to more than one SROCs. Based upon the received L1 triggers, each SROC searches for the corresponding L0 data in the associated VMM Capture channels. For each L1 trigger an L1 event is formed in each enabled SROC. This event contains the aggregated and reformatted associated valid L0 data. The L1 events are buffered into SRAMs implementing a FIFO strategy. As soon as they are ready to be sent (i.e. they are complete), they are 8b10b encoded, serialized and transmitted. Each SROC has two SLVS DDR serial output lines that can operate at up to 320 Mbps (i.e. 320×10^6 bps) each. Thus, the following configurations are possible: i) 80, 160 or 320 Mbps with only the first line active or ii) 640 Mbps with both lines operating at 320 Mbps and the transmitted bits being interleaved, same as for the VMM3 L0 data. The SROC should be able to saturate the output bandwidth in all four cases (i.e. achieve the maximum theoretical throughput as determined in Section 2.4). These output interfaces and the associated data formats are presented in sections 2.3.1 and 2.3.2, respectively. For the ROC to be considered real-time, the response of the SROC to every L1A command should be transmitted within a limited time frame. Thus, a watchdog timer with a configurable threshold of up to 12.8 μs should be used to limit the maximum time the SROC spends waiting for the required L0 data from an associated VMM Capture channel.

4. The ROC should be compatible with both one and two levels of hardware triggers. When only one trigger level is deployed, the VMM Capture channels will buffer the incoming L0 events from the interfaced VMM3s. At the same time, the ROC will receive L1A commands that match all these L0 events. If the L1A arrives before the corresponding L0 event is fully buffered within the VMM Capture FIFO then the SROC packet building FSM will wait for its arrival/completion (for a limited time, as explained above). The result is a 100% selection of the L0 events (i.e. assuming no data loss). When two levels of triggers are used, the SROC will discard the L0 data not matched by any L1 trigger and transmit only the selected data, thus acting as a filter.

5. Within a NSW TDAQ system that operates with only one level of hardware trigger, the ROC should handle a maximum L0A (and L1A) average rate of 1 MHz with a maximum fixed latency (i.e. the time between the selected BC and the arrival of the corresponding L0A/L1A command at the output of FELIX to the FEBs [177]) of 10 μs. When two trigger levels are used, the L0A remains as previously specified for the single-stage trigger while the L1A will have an average rate up to 1 MHz (but less than the L0A rate since not all BCs selected by the L0 trigger will also be selected by the L1 trigger) and a maximum fixed latency of 60 μs.

   The newer ATLAS Front-End Interface Requirements document [177] specifies the same L0A/L1A maximum rate and latency for the single-level trigger system as the VMM3-ROC joint requirements document. However, for the two-level trigger scheme, the stated maximum average L0A rate is 4 MHz in conjunction with the 10 μs latency. The L1A maximum average rate is decreased to 800 kHz and its latency to 35 μs. The new parameters are declared as final (i.e. not expected to be changed).

6. For debugging and calibration purposes, the ROC should implement a *bypass* option in which no matching between the L1 triggers and the L0 data is made. In this case, only one VMM Capture channel can be associated with each SROC so the aggregation function is also dropped. For each L1A command each SROC generates and transmits one output packet based on the oldest L0 packet from the associated VMM Capture FIFO. Basically, in this mode the SROC acts as a buffer that at the output also re-formats the data.

7. The ROC should implement Triple Modular Redundancy (TMR) to all the configuration registers and FSMs to mitigate the effects of SEUs (Single Event Upsets). An SEU pad and/or an SEU counter accessible to the SCA chip (through the configuration and register bank) should be implemented to perform radiation qualification tests and later on monitor the chip operation within ATLAS. In addition, parity bits, checksum fields and mechanisms for error identification and discarding should be implemented. These are described in sections 2.3.2 and 2.5.

8. The ROC logic should implement an active-low asynchronous full reset delivered through an input pin via the SCA chip.

9. The ROC should be compatible with the VMM3's congestion and flow control mechanisms. In addition, it should implement new mechanisms. All these are presented in sections 2.3.1, 2.3.2 and 2.3.3.

10. The configuration and monitoring of the ROC's packet processing logic should be achieved through a register bank interfaced with an Inter-Integrated Circuit (I^2C) slave. The slave is interfaced with the SCA ASIC. For the control of the ROC PLLs (Phase-Locked Loops) and the phase-shifting logic for the supplied clock and control signals, a separate I^2C slave and associated register bank should be implemented.

11. The ROC ASIC should be fabricated using the International Business Machines Corporation (IBM) 130 nm CMOS technology (now Global Foundries) on a Multi-Project Wafer (MPW) which also includes the VMM3 chip. The approx. number of pads is 200 while it is expected that the design will be pad limited (the larger number of pads will determine the die area, not the core logic) to a total area of approx. 20 mm^2. The recommended package is Ball-Grid Array (BGA) with a size between 12×12 mm^2 and 16×16 mm^2. The chip will be supplied with a nominal voltage of 1.2 V. The expected total draw current is approx. 300 mA (i.e. resulting in 0.36 W of power draw).

2.2 ROC context and top-level architecture

The ROC context within the NSW readout system, part of the NSW TDAQ system, is depicted in Figure 2.1. A top-level view of the ROC architecture is included. The ROC's logic is divided into two distinct parts with different purposes: the logic that supplies the required internal and external clock signals and the decoded TTC commands, with a light red background and being referred to as the *analog* part and the packet processing logic, depicted on a light yellow background and being

referred to as the *digital* part. The ROC's main data interfaces and clock signals are represented as connections to the other context ASICs.

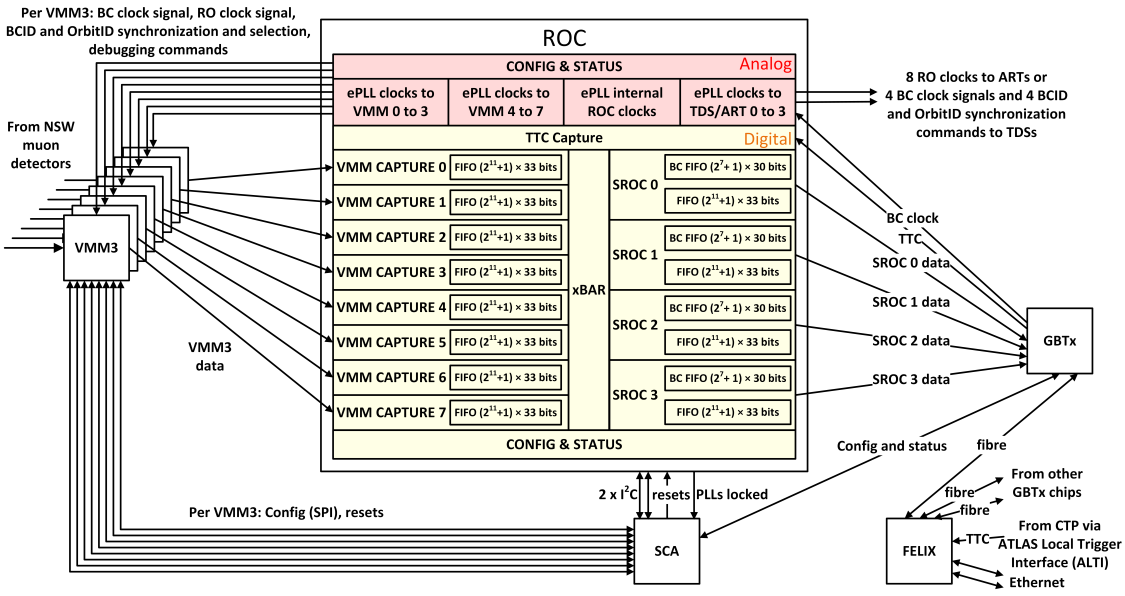


Figure 2.1: The ROC context within the NSW readout system, its main interfaces and top-level architecture.

The digitized data of up to 512 NSW detector channels are received and buffered by the ROC in its eight VMM Capture modules, each one being interfaced with a VMM3 ASIC. The TTC stream, distributed through the L1DDC GBTx, is interpreted by the ROC's TTC Capture module. In response to the L1A commands and the synchronized internal BCID and OrbitID counters, the selected BCs are buffered within each of the four SROCs in their BC FIFOs. The SROC modules process the associated L0 data based on these selections. The resulting L1 data is buffered and as soon as possible transmitted to the L1DDC GBTx. The association between the VMM Capture channels and the SROC modules is achieved through a configurable memoryless (i.e. purely combinational) cross-bar routing module. The ROC's digital Configuration and Status module is a register bank interfaced with the digital logic on one side and with an I^{2}C slave on the other. Some of these registers drive signals within the digital part (i.e. configuration) while the other registers supervise relevant signals from the digital logic (i.e. monitoring).

The ROC's internal and externally supplied clock signals are generated by four PLL[1] blocks [169] within the analog part. They all use the TTC BC clock signal as reference. The three ePLLs supplying the external clock signals are slightly modified versions of the design presented in [169]. They include phase-shifting circuits that receive the relevant TTC commands from the digital part (i.e. TTC Capture) and forward them with configurable phase. All the ePLLs are configured and monitored through a separate register bank associated with an I^{2}C slave.

The two ROC I^{2}C slaves (i.e. from the *digital* and *analog* parts) are interfaced to the master SCA chip on dedicated buses as shown in Figure 2.1. Several SCA GPIOs (General Purpose Input/Outputs) drive the ROC reset signals and sample

[1]called ePLL within the ATLAS collaboration as shown in [169].

its ePLLs locking signals. The SCA ASIC also configures the eight VMM3s through dedicated SPIs (Serial Peripheral Interfaces) and drives their reset signals.

One VMM3 ASIC manages 64 NSW detector channels and constructs and transmits L0 packets at 640 Mbps based upon the received L0A pulses (from ROC). One ROC manages the data from up to eight VMM3 ASICs resulting in up to four 640 Mbps L1 data streams. One L1DDC GBTx aggregates the data from multiple SROCs until its output bandwidth (up to 4.48 Gb/s effective) is saturated. While the VMM3 and ROC have filtering capabilities, the L1DDC GBTx multiplexes the data from all the associated SROCs as chunks of the output streams, alongside the SCA data. It also provides the reference 40 MHz BC clock and the 320 Mbps TTC stream to the ROC and forwards the configuration commands to the SCA ASIC. Thus, the L1DDC GBTx acts as the interface between the NSW readout system and the FELIX system.

The entire digital part of the NSW readout system is a chain of multiplexers and/or de-multiplexers with memory (i.e. buffering spaces) that are managed by FSMs coordinated through control interfaces (e.g. the TTC stream). The control commands are generated by the triggering path. As the downstream data path goes from the front-end to the back-end, the used output interfaces have higher and higher bandwidths while aggregating data from more and more sources (i.e. detectors channels). For the upstream path (i.e. from the back-end to the front-end; e.g. configuration, TTC streams) the reverse is valid.

2.3 ROC architecture

The main data flows and clock signals within the ROC top-level architecture are depicted in Figure 2.2. There is no separation between the *digital* and *analog* parts. The *analog* part consists of the *ePLLs*, *ePLL* and *ePLL Config* blocks. The rest represents the *digital* part. The *ePLLs* block is formed by the three modified ePLLs that supply the external clock signals and the TTC commands from TTC Capture using digital phase-shifting circuits. Each block from the digital part is detailed in this section. The digital part modules are supplied with TMR BC and RO clock signals. Details about the ROC TMR are presented in Chapter 4, Section 4.2.1. In this chapter, the presented block diagrams, schematics and tables related to the ROC's architecture do not depict or describe the TMR logic or its signals (except the Config module). They are intended to present the ROC's logic functionality. The BC-synchronous SR TTC command supplied by the TTC Capture to all the other *digital* modules is also not depicted or described for simplicity. It implements the functionality described in Section 2.1.

The internal architecture of the VMM Capture channel is depicted in Figure 2.3. The VMM3 serial data stream (i.e. *serial_data_i*) is first deserialized into 10-bit wide words (i.e. the *enc_data* bus validated by the *data_valid_des* signal). Then, the *Comma Align* module determines the correct 8b10b stream alignment, as described in Section 2.3.1. The resulting 8b10b symbols (i.e. the 10-bit *enc_aligned_data* bus validated by the *aligned* and *data_valid_align* signals) are decoded and fed to the *Assembler* module through the 8-bit *dec_data* bus and the associated *comma* signal, both validated by the *dec_data_valid* signal. The *Assembler* module checks the protocol and the data integrity and pushes the resulting valid L0 packets into the FIFO, through the 33-bit wide *assembl_data* bus validated by the *fifo_wr* signal. The

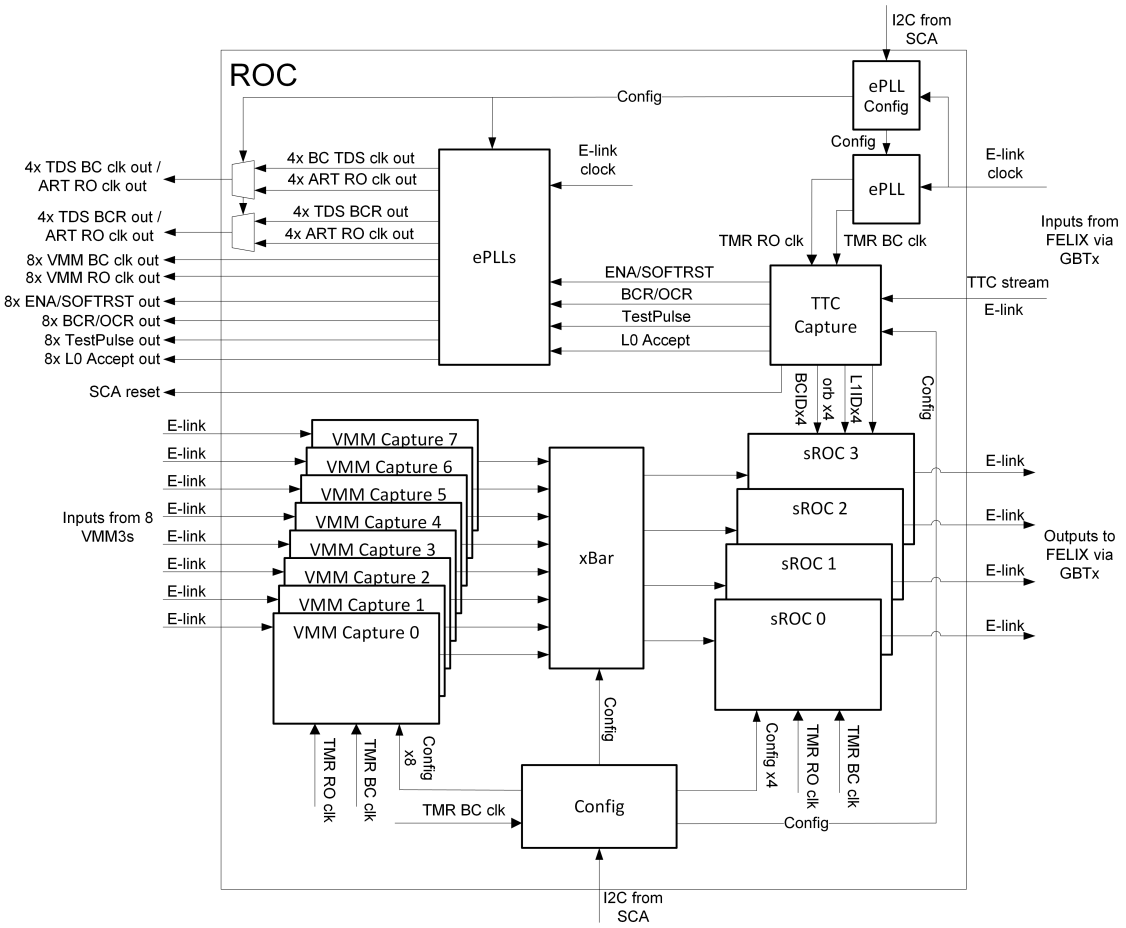


Figure 2.2: The top-level ROC architecture showing the main clock and data signals. From [79] updated and corrected.

chosen bus width is motivated by the data formats which are presented in Section 2.3.2. If only one address is free in the FIFO (i.e. signaled through *fifo_almost_full*), then the current packet is truncated. The logic of the VMM Capture module is paced by the internal RO clock signals. The VMM Capture FIFO also passes the input data into the BC clock domain. The VMM Capture status signals are detailed in Table 2.1. The writing of the L0 data can be disabled using the *fake_vmm_failure* signal (from Config). The parity check can be set to be even or odd using the *even_parity* signal (from Config).

In Figure 2.4 additional logic from within the VMM Capture is detailed. The five 6-bit wide registers at the bottom cross the status signals described in Table 2.1 from the launching RO clock domain to the capturing BC clock domain of the Config module. The two almost full FIFO signals have configurable thresholds (i.e. from the Config block) and are used to signal to the associated SROC when to send corresponding flow control signals in the output data stream (see sections 2.3.1 and 2.3.3). The rest of the depicted logic is responsible for the automatic disabling feature of the VMM Capture channel, called timeout. Each VMM Capture contains a watchdog timer (i.e. the COUNTER) with a configurable threshold (i.e. *timeout_i*) that increments when this feature is enabled in the Config block (i.e. *timeout_en_i* is high), the FIFO is empty (i.e. no valid L0 data is received) and at least one L1 trigger was received. This watchdog timer is cleared when the VMM Capture

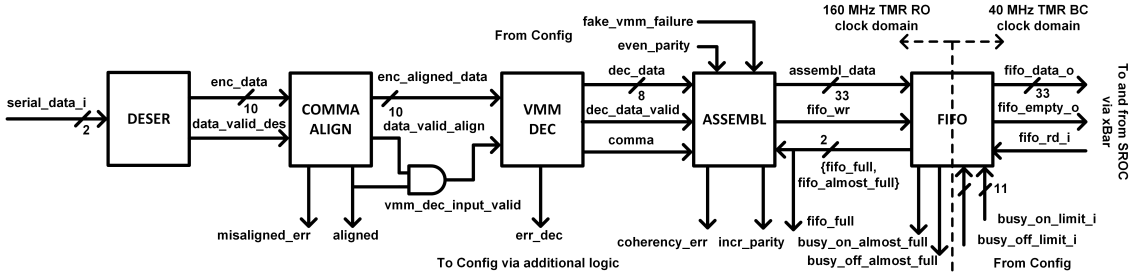


Figure 2.3: The internal architecture of a VMM Capture channel showing the evolution of the main data buses that link its components. From [79] updated.

Signal name	**Description**
fifo_full	The VMM Capture FIFO is full.
coherency_error	The L0 packet is not complete.
err_dec	8b10b decoding error.
misaligned_err	The alignment of the input 8b10b stream changed.
aligned	The alignment to the input 8b10b is achieved.
incr_parity	Indicates an L0 packet with incorrect header parity.

Table 2.1: The descriptions of the 1-bit wide status signals supplied by the VMM Capture channel (Figure 2.3) to the Config module (Figure 2.9).

FIFO empty signal is de-asserted. If however the threshold is reached the channel is automatically disabled and the situation is signaled to the Config block through *timeout_flag_o*). Within Config a flag is set (i.e. *timeout_status_bit_i*). Only when this flag is cleared the VMM Capture channel is re-enabled.

The block diagram of the intermediate configurable cross-bar (xBar) module that links the eight VMM Capture channels to the four SROC modules is presented in Figure 2.5. Its internal organization is detailed in Figure 2.6, where for the generation of the *fifo_read_o* signals only one case is depicted (the others are similar). The module is a memoryless (i.e. purely combinational) switching fabric between the input and output channels consisting of multiplexer and demultiplexer circuits. The VMM Capture FIFOs read domain data and control signals are routed to the SROCs. Thus, the routing of signals is bi-directional (i.e. signals are going from the VMM Capture FIFO to the SROC and vice-versa). The selection commands are dictated by the four SROCs through the *vmm_sel_i* signals. The *vmm_ena_i* bus is sourced from the Config module and is formed by concatenating the four 8-bit wide SROC-VMM Capture association lists. E.g. if SROC 0 is responsible for VMM Capture channels 0, 1, 3 and 7, then its association list is *0b1000_1011*. Within the cross-bar these lists are used to enable the issuing of the read commands. The only rule is that one VMM Capture cannot be associated with more than one SROC.

The internal architecture of the SROC module is depicted in Figure 2.7 and its IO signals are described in Table 2.2. The ROC core/main data processing algorithm, presented in Section 2.5, is implemented as the Packet Builder FSM. The TTC FIFO (also called BC FIFO) buffers the L1 triggers between the TTC Capture module and the Packet Builder FSM. The Packet Builder FSM controls the cross-bar and

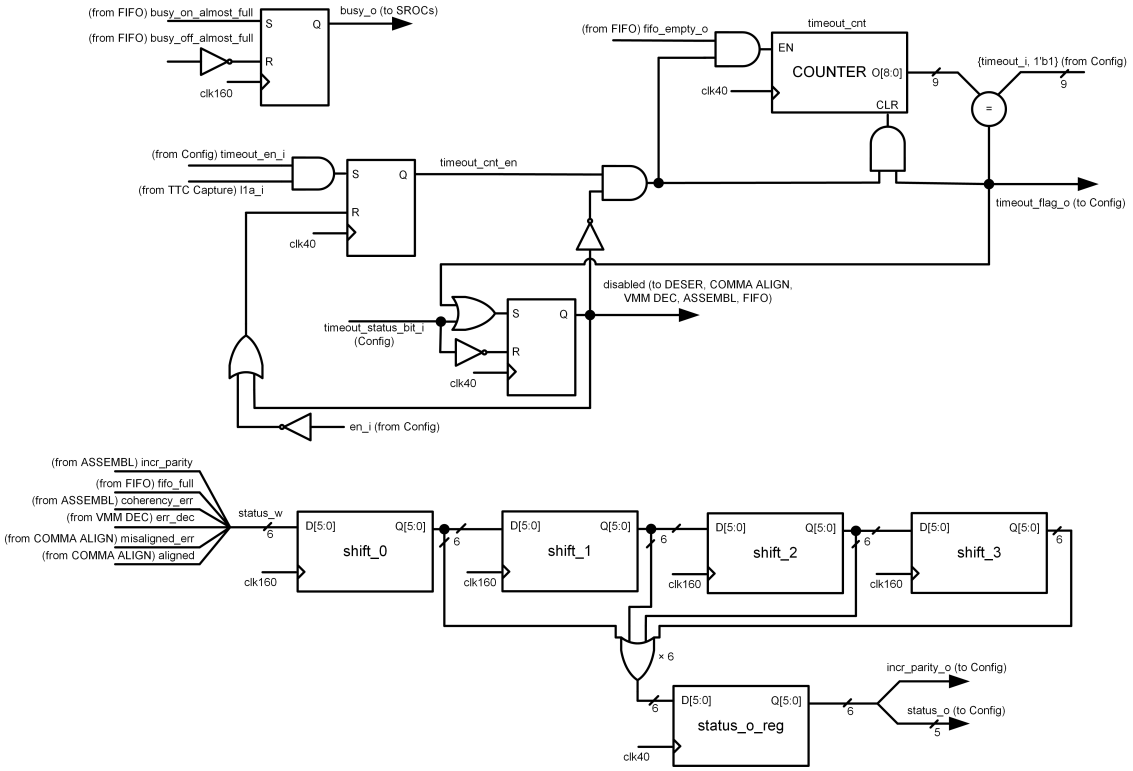


Figure 2.4: Detailed logic of the VMM Capture channel complementary to what is depicted in Figure 2.3. From [7] updated.

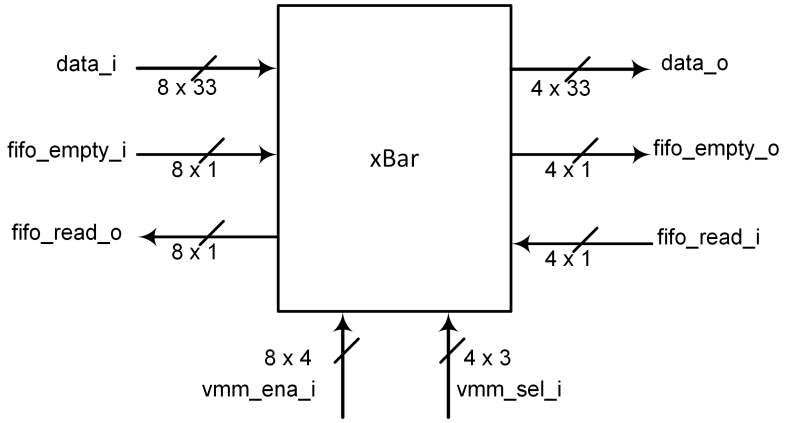


Figure 2.5: The block diagram of the cross-bar module that interfaces the eight VMM Capture modules with the four SROCs. From [7] updated.

implements the majority of the packet processing specifications. In response to the L1 triggers it checks, aggregates, re-formats and filters the L0 packets from the associated VMM Capture modules, building output packets that are pushed into the Packet FIFO (also called SROC FIFO). Basically, each output packet contains the data from all the input packets matched by a trigger. The Streamer FSM pops the L1 data from the Packet FIFO, generates the output data stream and supplies it to the 8b10b encoder byte by byte. It implements the output protocol described in Section 2.3.1 and through its output data validation signal (i.e. *streamer_data_valid*) it establishes the cadence of the data transfer for the downstream modules. The 8b10b encoded data is sent by the Feeder module only to the first serializer (for the

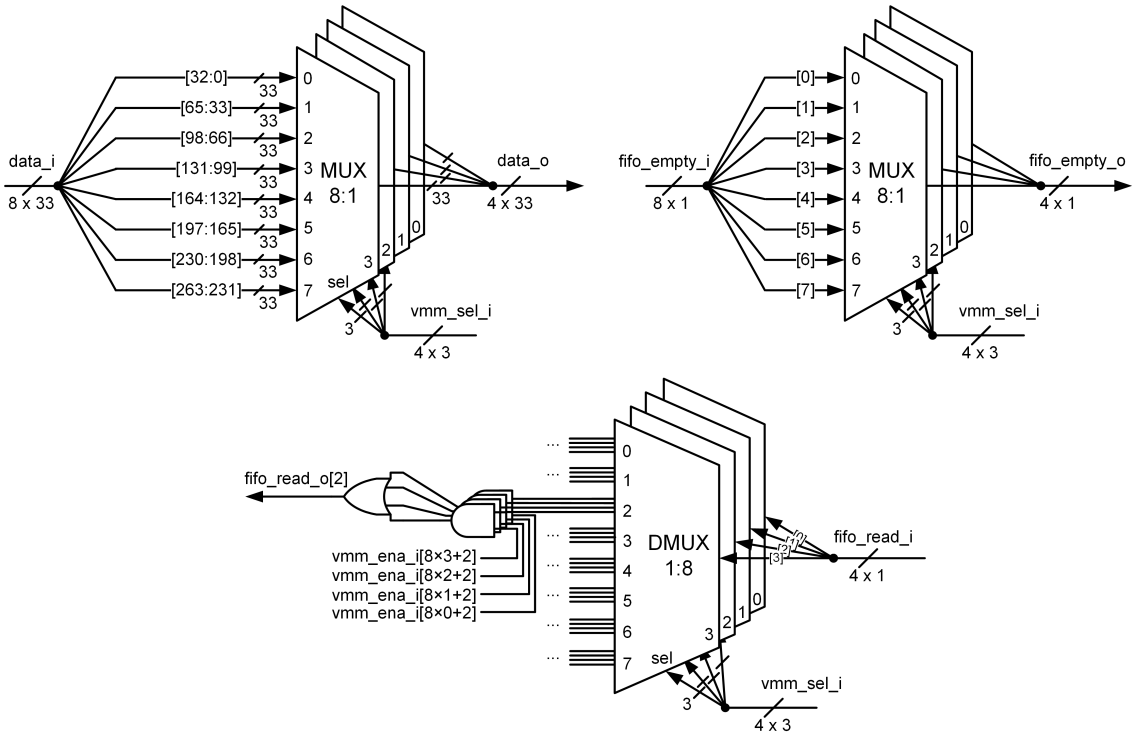


Figure 2.6: The organization of the ROC cross-bar module. Only one case is depicted for the generation of the *fifo_read_o* signals.

80, 160 and 320 Mbps speeds) or with the bits interleaved to both serializers (for the 640 Mbps speed).

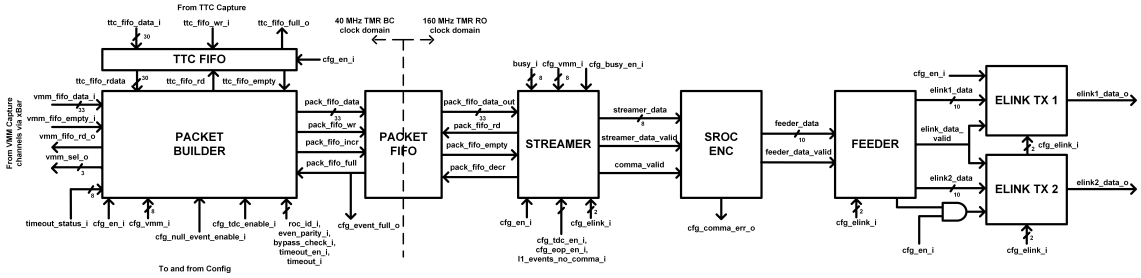


Figure 2.7: The internal architecture of an SROC channel showing its main components and the evolution of the main data buses that link them. From [79] updated.

If the timeout functionality is enabled (i.e. *timeout_en_i* is high) then the Packet Builder FSM will skip the associated VMM Capture channels that were automatically disabled by the absence of L0 data (signaled through *timeout_status_i*). If however the timeout feature is disabled then the corresponding logic from Figure 2.4 is disabled and the timeout flags remain cleared regardless of the presence of L0 data. In this case, the Packet Builder FSM implements a watchdog timer so that it will wait for maximum *timeout_i* $\times$ 2 BCs at each empty VMM Capture channel.

The Packet Builder FSM and the TTC FIFO are paced by the internal BC clock signals while the rest of the SROC logic uses the RO clock signals. Thus, the SROC FIFO also transfers the output data between these clock domains.

Signal name	Description
	VMM Capture FIFO interface via xBar
vmm_fifo_data_i	Read data bus from the selected VMM Capture FIFO routed through the cross-bar.
vmm_fifo_empty_i	Empty signal from the selected VMM Capture FIFO routed through the cross-bar.
vmm_fifo_rd_o	Read signal to the selected VMM Capture FIFO routed through the cross-bar.
vmm_sel_o	The VMM Capture selection command that controls the cross-bar multiplexers and de-multiplexers.
	TTC FIFO interface
ttc_fifo_data_i	Write data bus from the TTC Capture to the TTC FIFO.
ttc_fifo_wr_i	Write signal from the TTC Capture to the TTC FIFO that validates the data present on *ttc_fifo_data_i*.
ttc_fifo_full_o	Full signal of the TTC FIFO to the Config and TTC Capture modules.
	Configuration and status signals from and to the Config module
cfg_en_i	Enable signal for the SROC module.
cfg_elink_i	The speed for the serializers (*0b00* is 640 Mbps, *0b01* is 320 Mbps, *0b10* is 160 Mbps and *0b11* is 80 Mbps).
cfg_vmm_i	Specifies the associated VMM Capture channels. Based upon this signal, the Packet Builder FSM generates the *vmm_sel_o* command to the cross-bar.
cfg_null_event_enable_i	Enable signal for the transmission of L1 null events. See Section 2.3.2.
cfg_tdc_en_i	Enables the transmission of the Time to Digital Converter (TDC) measure in the output hit words. See Section 2.3.2.
cfg_eop_en_i	Enables the transmission of the End Of Packet (EOP) symbols between back to back L1 packets. See Section 2.3.1.
cfg_busy_en_i	Enables the transmission of Busy-On / Busy-Off control symbols if applicable. See Section 2.3.1.
roc_id_i	The 6-bit wide ROC IDentifier (ID). See Section 2.3.2
even_parity_i	Indicates the used parity (1 is even, 0 is odd).
bypass_check_i	Indicates the operation mode for the Packet Builder FSM. See Section 2.5.
timeout_en_i	Indicates if the timeout feature is enabled.
timeout_i	The timeout watchdog threshold.
timeout_status_i	The timeout flags of all the VMM Capture modules.
l1_events_no_comma_i	The threshold for the maximum number of back-to-back L1 packets sent without any *comma* symbols. See Section 2.3.1.
cfg_event_full	Indicates that the Packet FIFO is full.
cfg_comma_err_o	Indicates that an 8b10b encoding error has occurred.

E-link serial interface	
elink1_data_o	The serial output to the first E-link.
elink2_data_o	The serial output of the second E-link which is used only when the transmission speed is 640 Mbps.
Other	
busy_i	The busy state for each VMM Capture FIFO. See Section 2.3.3.

Table 2.2: The descriptions of the SROC IO signals.

The TTC Capture module, with the block diagram presented in Figure 2.8 and the interfaces detailed in Table 2.3, processes the TTC stream. It implements the ROC BCID, OrbitID and L1ID counters, executes the TTC commands relevant to the ROC and supplies the ones necessary for the associated VMM3 and TDS ASICs to the phase aligners within the analog ROC part. The byte alignment of the input stream is determined by detecting the positive edge of the BC clock signal in the RO clock domain (more details in Section 2.3.1). For the ROC-relevant TTC commands, optional parallel inputs interfaced directly with ROC pads are included. To increase the observability, some of them these pads can be configured as outputs being driven by the corresponding deserialized bits in this case. These additions to the module's debuggability were crucial during the ROC design validation and testing, as presented in Chapter 3.

The L1A commands push the current values of the internal BCID, OrbitID and L1ID counters into the TTC FIFO of each SROC, as long as they are not full.

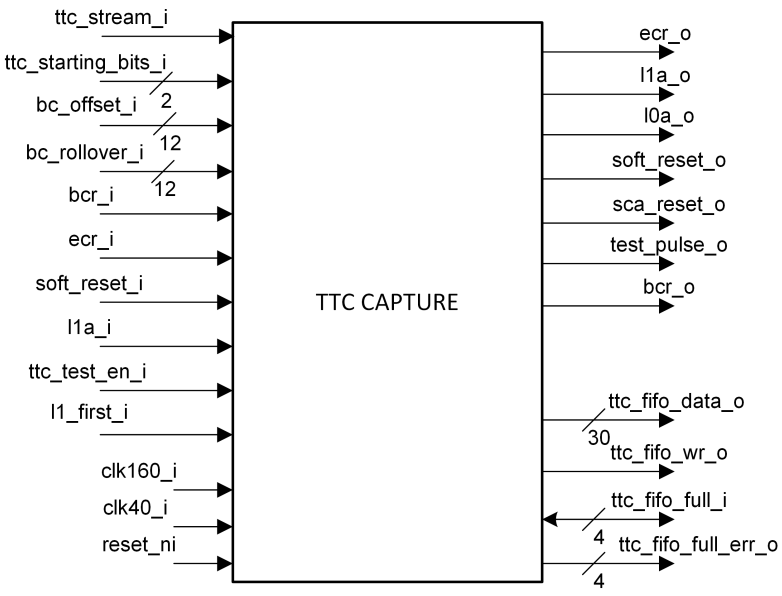


Figure 2.8: The block diagram of the TTC Capture module which decodes the TTC stream, forwards the resulted TTC commands to the analog part to be phase-adjusted and distributed in the ROC context and pushes the detected L1 triggers in the TTC/BC FIFO of each SROC. From [7] updated.

Signal name	Description
ttc_stream_i	The serial DDR data input representing the TTC stream, fed directly from the corresponding ROC input pad.
Configuration and status signals from and to the Config module	
ttc_starting_bits_i	Selects from pre-defined alignment and bits order scenarios. See Section 2.3.1.
bc_offset_i	The offset value for the internal BCID counter to be loaded upon the receipt of a set BCR bit.
bc_rollover_i	The rollover value for the internal BCID counter.
ttc_test_en_i	Signal driven by a ROC input pad that enables the parallel fed debug ROC-relevant TTC commands (i.e. *bcr_i*, *ecr_i*, *soft_reset_i* and *l1a_i*), bypassing the *ttc_stream_i*.
l1_first_i	Dictates the start value for the L1ID counter (0 or 1).
ttc_fifo_full_err_o	For each SROC indicates when a push operation into the corresponding TTC FIFO was discarded due to the FIFO being full.
Parallel fed input and output debug TTC signals	
bcr_i	Input BCR signal driven by a dedicated ROC input pad.
ecr_i	Input ECR signal driven by a dedicated but bidirectional ROC pad. Thus it cannot be used simultaneously with *ecr_o*. The direction of this pad is controlled by a separate signal driven by a ROC pad called *test_highz*.
soft_reset_i	Input SR signal driven by a dedicated ROC input pad.
l1a_i	Input L1A signal driven by a dedicated but bidirectional ROC pad. Thus it cannot be used simultaneously with *l1a_o*. The direction of this pad is controlled by a separate signal driven by a ROC pad called *test_highz*.
l1a_o	Output L1A signal supplied to the bidirectional ROC pad that alternatively supplies *l1a_i*. The direction of this pad is controlled by a separate signal driven by a ROC pad called *test_highz*.
ecr_o	Output ECR signal supplied to the bidirectional ROC pad the alternatively supplies *ecr_i*. The direction of this pad is controlled by a separate signal driven by a ROC pad called *test_highz*.
Output TTC commands synchronous to the BC clock signal supplied to the ROC's digital logic, ROC's pads and/or to the phase aligners for external distribution	
soft_reset_o	The resulted SR TTC command supplied to the ROC's digital part and the associated VMM3's via the analog part.
l0a_o	The resulted L0A TTC command to be supplied to the associated VMM3s via the ROC's analog part.
sca_reset_o	The resulted SCA reset TTC command to be supplied directly to the SCA ASIC via an output pad.
test_pulse_o	The resulted TP TTC command to be supplied to the associated VMM3s via the ROC's analog part.
bcr_o	The resulted BCR TTC command to be supplied to the associated VMM3 and TDS ASICs via the ROC's analog part.

TTC FIFOs interface	
ttc_fifo_data_o	Data bus through which the L1 triggers are being pushed.
ttc_fifo_wr_o	Write signal that validates the data present on *ttc_fifo_data_o*.
ttc_fifo_full_i	Concatenated TTC FIFO full signals from the four SROCs.

Table 2.3: The descriptions of the TTC Capture IO signals depicted in Figure 2.8.

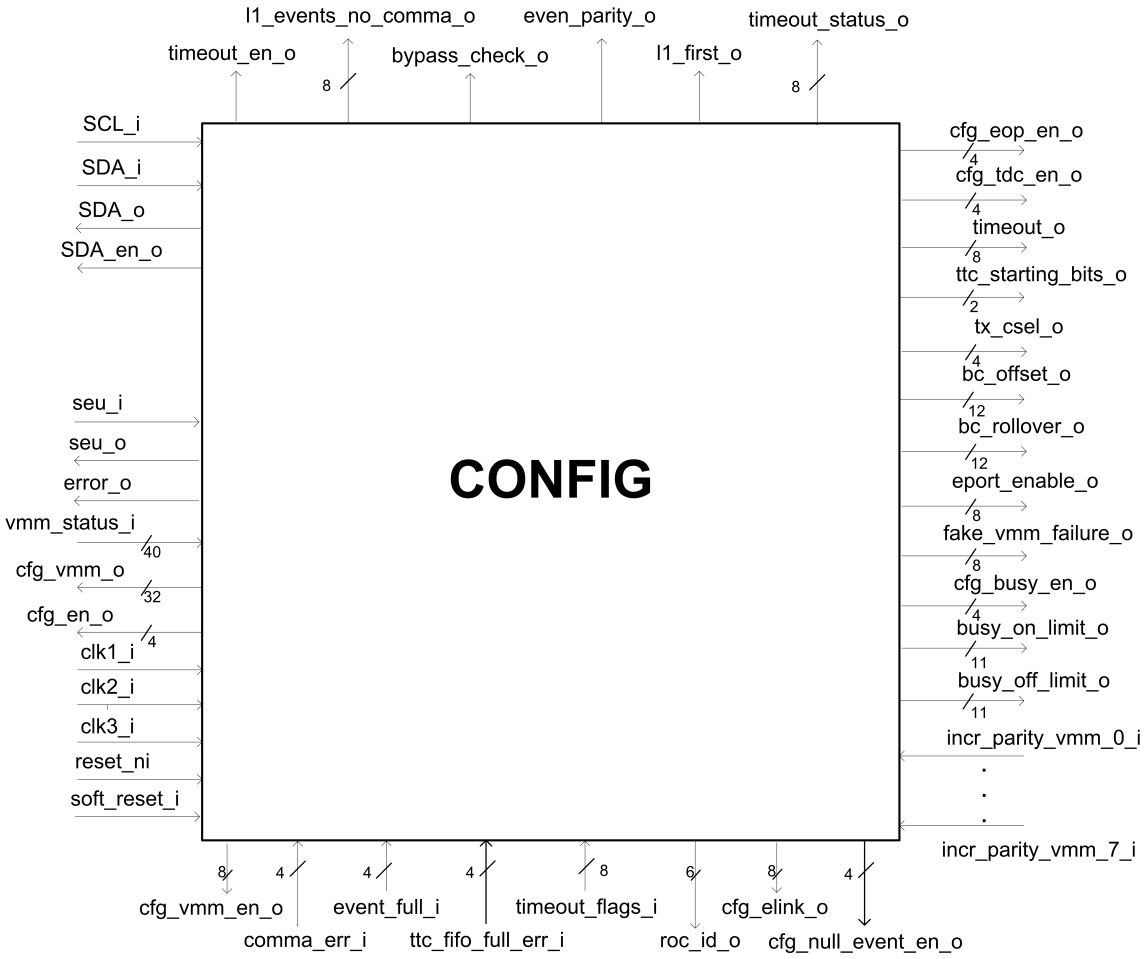


Figure 2.9: The block diagram of the Configuration module for the digital part [7].

Both the digital and analog parts contain dedicated Configuration and Status modules consisting of an I^2C slave that translates the received transactions into read and write commands addressed to an associated register bank. The block diagram of the Config module for the digital part is depicted in Figure 2.9. Some of the registers within the bank are used to configure various settings for all the other digital part modules (e.g. association lists for the four SROCs, transmission speeds, etc.). They drive the output signals from Figure 2.9 which are also detailed in Table 2.4 as configuration. These signals are meant to be configured and then an SR TTC command to be issued. The SR resets the rest of ROC's digital logic. Upon the release of the SR the logic starts directly with the set configuration. However, the SR is not mandatory. Depending on the changes and the state of the ROC's digital logic at that moment, protocol and data formats errors might be caused, hence the

SR necessity. The I^2C interface is relatively slow compared to the BC clock signal that paces them. The exact moment of the value change relative to the moment of the I^2C transaction issue is not constrained. Nevertheless, during the synthesis and implementation ASIC design stages, these registers were constrained to change the value in a single BC clock cycle (i.e. no multicycle path exception) once the write I^2C transaction provided all the necessary data. The default values (i.e. values after reset) for all the ROC digital configuration signals are mentioned in Table 2.4.

The other registers are influenced by different signals supplied from all the other digital part modules. These signals are detailed in Table 2.4 as status monitors. They represent relevant values and states reached by the digital modules (e.g. 8b10b alignment errors, full FIFOs, etc.). The corresponding registers act as flags or counters set or incremented at each high pulse from the source. Upon a read I^2C transaction, their value is sampled, transmitted and then cleared. The only exception is the VMM Capture's *aligned* bit that indicates the current state of alignment. The same constraints as for the configuration registers are used.

Similarly, within the ROC's analog part, the ePLL Config module contains registers controlling and monitoring the ePLLs settings and conditions, respectively.

All the ROC FIFOs are based upon the model described in [90], use SRAM and registers as buffer space and are First Word Fall Through (FWFT). The resulting address space size is in the $2^n + 1$ format, where n is the number of bits used to address the SRAM. Except for the TTC/BC FIFOs, the FIFOs are asynchronous being used to also transfer data between the RO and the BC clock domains. In Figure 2.1, each FIFO has its total buffer size listed in the *depth* $\times$ *width* format.

Signal name	**Description**
	I^2C interface
SCL_i	The Serial Clock Line (SCL).
SDA_i	The input Serial Data Line (SDA).
SDA_o	The output SDA.
SDAen_o	When high the IO pad corresponding to the SDA line is output driven by *SDA_o*, else it is input that drives *SDA_i*. It controls the tri-state buffer within the SDA pad.
	Configuration (i.e. output signals)
cfg_vmm_o	8-bit VMM Capture - SROC association list per SROC. Connected to *cfg_vmm_i* from Table 2.2 and Figure 2.7. By default SROC 0 receives data from VMM Capture channels 0 and 7 (i.e. *cfg_vmm_o[7:0]* = 0x81); SROC 1 from channels 1 and 6 (i.e. *cfg_vmm_o[15:8]* = 0x42); SROC 2 from channels 2 and 5 (i.e. *cfg_vmm_o[23:16]* = 0x24) and SROC 3 from channels 3 and 4 (i.e. *cfg_vmm_o[31:24]* = 0x18).
cfg_en_o	Per SROC enable signal. Connected to *cfg_en_i* from Table 2.2 and Figure 2.7. By default all the SROCs are enabled (i.e. 0xF).
cfg_vmm_en_o	Per VMM Capture enable signal. Not depicted in Figures 2.3 and 2.4. By default all the VMM Capture channels are enabled (i.e. 0xFF).
cfg_elink_o	Per SROC Elink speed. Connected to *cfg_elink_i* from Table 2.2 and Figure 2.7. By default all four SROCs are set at the maximum 640 Mbps speed (i.e. 0x00)

roc_id_o	6-bit wide ROC ID used in some output packets. Connected to *roc_id_i* from Table 2.2 and Figure 2.7. See Section 2.3.2. By default it is 0.
cfg_null_event_en_o	Per SROC enable signal for the sending of null event packets. Connected to *cfg_null_event_enable_i* from Table 2.2 and Figure 2.7. See Section 2.3.2. By default it is enabled for all SROCs (i.e. 0xF).
cfg_eop_en_o	Per SROC enable signal for the sending of End Of Packet (EOP) symbols between back-to-back output packets. Connected to *cfg_eop_en_i* from Table 2.2 and Figure 2.7. See Section 2.3.1. By default it is enabled for all the SROCs (i.e. 0xF).
cfg_tdc_en_o	Per SROC enable signal for the sending of the Time to Digital Converter (TDC) measure in the output packets. Connected to *cfg_tdc_en_i* from Table 2.2 and Figure 2.7. See Section 2.3.2. By default it is enabled for all the SROCs (i.e. 0xF).
ttc_starting_bits_o	Selects the alignment and bit order scenario for the TTC commands within the TTC byte. Connected to *ttc_starting_bits_i* from Table 2.3 and Figure 2.8. See Section 2.3.1. By default it is 0.
timeout_o	The timeout threshold for the VMM Capture and SROC watchdog timers. Connected to *timeout_i* from Figures 2.4 and 2.7. By default it is set to the maximum (i.e. 0xFF equivalent to 12.8 μs).
tx_csel_o	Setting for output current of the tx e-link [62] pads. See Section 2.3.5. By default set to 2 mA as for the SLVS standard.
bc_offset_o	The value to be loaded into the BCID counter upon the receipt of a BCR TTC command. Connected to *bc_offset_i* from Table 2.3 and Figure 2.8. By default it is set to 1157 (i.e. 60 μs before rollover).
bc_rollover_o	The rollover value for the BCID counter. Connected to *bc_rollover_i* from Table 2.3 and Figure 2.8. By default it is set to 3557.
eport_enable_o	Individual enable signals for the tx e-link [62] pads associated with SROC serial data outputs. See Section 2.3.5. By default all the SROC tx e-links are enabled.
fake_vmm_failure_o	Per VMM Capture signal that disables the pushing of the received L0 data within the VMM Capture FIFO. Connected to *fake_vmm_failure* from Figure 2.3. By default the pushing of data within the FIFO is enabled for all VMM Capture channels (i.e. 0).
cfg_busy_en_o	Per SROC signal that enables the sending of Busy-On and Busy-Off symbols in the output data stream, based upon the condition of the associated VMM Capture FIFOs (see Figure 2.4). Connected to *cfg_busy_en_i* from Table 2.2 and Figure 2.7. See sections 2.3.1 and 2.3.3. By default it is enabled for all the SROCs (i.e. 0xF).
busy_on_limit_o	The upper threshold for the VMM Capture FIFO's occupancy level that once crossed triggers the sending of a Busy-On control symbol in the output stream of the corresponding SROC module, if enabled (i.e. *cfg_busy_en_o*). Connected to *busy_on_limit_i* from Figure 2.3. See sections 2.3.1 and 2.3.3. By default it is set to 1900.
busy_off_limit_o	The lower threshold for the VMM Capture FIFO's occupancy level that once crossed after a Busy-On was issued triggers the sending of a Busy-Off control symbol in the output stream of the corresponding SROC module, if enabled (i.e. *cfg_busy_en_o*). Connected to *busy_off_limit_i* from Figure 2.3. See sections 2.3.1 and 2.3.3. By default it is set to 1800.
l1_first_o	The starting value (0 or 1) of the L1ID counter from TTC Capture. Connected to *l1_first_i* in Table 2.3 and Figure 2.8. By default it is 0.

bypass_check_o	Dictates the operation mode of the SROC's Packet Builder FSM. Basically if enabled it puts the SROC in a pass-through mode in which no matching of the L1 triggers with the L0 data is performed. Connected to *bypass_check_i* from Table 2.2 and Figure 2.7. See Section 2.5. By default this mode is disabled (i.e. 0).
timeout_en_o	Enables the timeout functionality. Connected to *timeout_en_i* from Table 2.2 and Figures 2.4 and 2.7. By default it is disabled (i.e. 0).
l1_events_no_comma_o	The maximum number of output packets that can be sent back-to-back. Connected to *l1_events_no_comma_i* from Table 2.2 and Figure 2.7. See Section 2.3.1. By default it is set to the maximum value which is 255.
even_parity_o	Dictates the used parity (even or odd). Connects to *even_parity* from Figure 2.3 and *even_parity_i* from Table 2.2 and Figure 2.7. See sections 2.3.2 and 2.5. By default an even parity is used (i.e. 1).
timeout_status_o	Timeout status flags set by the corresponding logic from Figure 2.4 through *timeout_flags_i*, only when the timeout functionality is enabled (i.e. *timeout_en_o* is high). Cleared upon explicit I^2C write transaction. Connected to *timeout_status_bit_i* from Figure 2.4 and *timeout_status_i* from Table 2.2 and Figure 2.7. By default it is set no VMM Capture channel is disabled (i.e. 0).
Status monitoring (i.e. input signals)	
vmm_status_i	Per VMM Capture 5-bit wide status signal. Depicted as *status_o* in Figure 2.4. From the Most Significant Bit (MSB) to the Least Significant Bit (LSB): *fifo_full*, *coherency_err*, *err_dec*, *misaligned_err* and *aligned*.
comma_err_i	Per SROC signal that indicates 8b10b encoding errors.
event_full_i	Per SROC signal that indicates that the Packet/SROC FIFO is full.
ttc_fifo_full_err_i	Per SROC signal that indicates that the TTC Capture module tried to push an L1 trigger into the full TTC/BC FIFO.
incr_parity_vmm_x_i, $\forall x \in \mathbb{N}, 0 \leqslant x \leqslant 7$	Indicates an L0 header with incorrect parity within VMM Capture channel x. The entire packet is dropped (i.e. is not pushed within the FIFO). Each signal increments a dedicated 8-bit counter that can be read through the I^2C interface.
timeout_flags_i	Timeout signals from the eight VMM Capture channels. Driven by *timeout_flag_o* from Figure 2.4.
seu_i	Indicates that an SEU has occurred in the ROC's digital logic outside the Config block. Sets a flag and increments an 8-bit SEU counter, both readable through the I^2C interface. See Chapter 4, Section 4.2.1.
Others	
error_o	Output signal that drives a ROC output pad and is set when any VMM Capture parity error counter is not cleared or any VMM Capture status signal is set (except *aligned*) or any SROC status bit is set (i.e. 8b10b encoding error or any FIFO full).
seu_o	Indicates that an SEU occurred within the Config module logic. An OR gate having the two SEU signals as inputs drives the ROC's output SEU pad. See Chapter 4, Section 4.2.1.

Table 2.4: The descriptions of the ROC's digital Config's IO signals depicted in Figure 2.9.

2.3.1 ROC interfaces

Understanding the ROC interfaces is essential for constructing its mathematical model, assessing its performance, understanding its architecture and organization and the design of its test environment (Chapter 3). In addition to the main data and control interfaces, various debug ports were implemented to increase the debuggability (i.e. controllability and observability) of the design. The number of these debug ports was limited by the ROC silicon die layout detailed in Section 2.3.5.

In Figure 2.10 the waveforms depict the four modes of transmission of 8b10b encoded data between the NSW FEBs and the L1DDC GBTx ASICs. The bits are tagged with one letter indicating the position within the 10-bit wide symbol and one digit representing the symbol number. In the *abcde**i**fgh**j*** symbol notation, *a* is the LSB and ***j*** the MSB. Bits ***i*** and ***j*** result from the 5b/6b and 3b/4b [199] encodings, respectively. The used electrical interface is called e-link, it complies with the SLVS standard [54], is radiation-hardened and can cope with data rates up to 320 Mbps (i.e. 320×10^6 bps) [62]. Thus, in the 640 Mbps mode, two 320 Mbps e-links are associated, the bits being interleaved. In this case, the 8b10b property of no more than five consecutive bits with the same value, the DC-balance[2] (Direct Current) and the running disparity[3] are not guaranteed for each individual e-link. For the other speeds, only one e-link line is used which maintains these properties. All four modes are available for the transmission of ROC's L1 data while only the 640 Mbps mode is used for the transmission of L0 data from VMM3 to ROC.

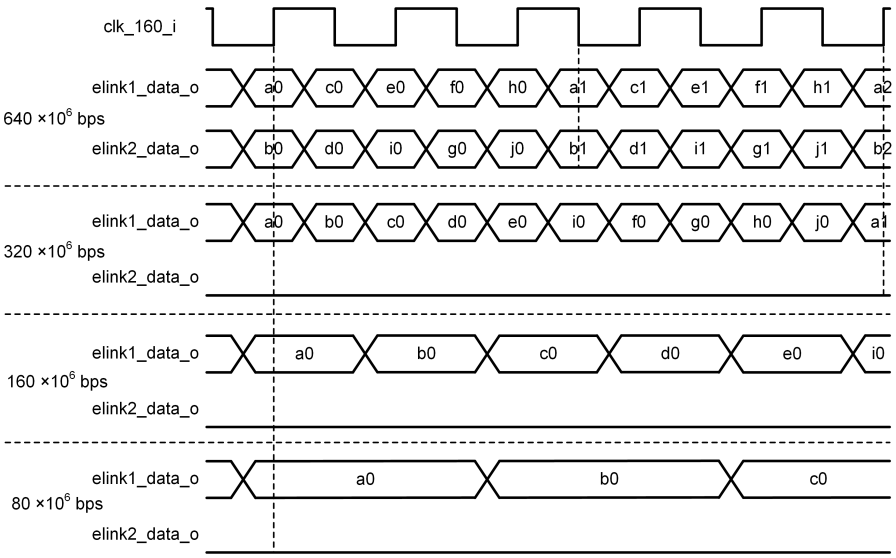


Figure 2.10: Waveforms depicting the four e-link [62] modes of operation for transmitting 8b10b encoded data between the NSW FEBs and the L1DDC GBTx ASIC.

The VMM3 ASIC transmits its L0 data 8b10b encoded on the two e-link serial lines at DDR on the 160 MHz RO clock signal supplied by the ROC. As shown in the 640 Mbps case from Figure 2.10, the bits are interleaved on the two lines. Thus, for transmitting one 10-bit symbol (i.e. one encoded byte) 2.5 cycles at a 160 MHz clock frequency are necessary. The resulting uncoded throughput is 512

[2]The DC component (i.e. the average amplitude) of a signal is zero. Prevents bit errors while passing through coupling capacitors.

[3]The difference between the number of transmitted 1 and 0 bits is always limited to ±2.

Mbps. The data is organized into packets, one for each L0 trigger (i.e. assuming that no trigger overflow happens inside the VMM3), which are separated by at least two special 8b10b symbols (i.e. the two possible encodings of the *K.28.5* special symbol), called *commas*. The *commas* have unique encodings that cannot be found by the concatenation of any two valid 8b10b symbols. Thus, they are used at the receiving end to determine the start position of the 8b10b symbols within the data stream. This process is called alignment or synchronization and is performed whenever a *comma* is received. The VMM3 packets do not contain special start and/or stop symbols. The first non-comma symbol after a *comma* represents the first byte (i.e. the most significant) of the packet while the last non-comma symbol before a *comma* represents the last byte of the packet (i.e. the least significant).

Each SROC has two output e-links (see Figure 2.7) and can be individually configured to transmit its L1 data at any of the four modes of operation (see *cfg_elink_o* from Table 2.4). The ROC output data is also organized into packets, one for each L1 trigger (i.e. assuming that no trigger overflow happens inside the ROC). The same *commas* as in the VMM3 case fill the data stream when packet data is not available for transmission. The L1 packets can be transmitted back-to-back but periodically two *commas* are inserted into the data stream for the FELIX to maintain the alignment (the GBTx is transparent to the E-link bit stream). Start Of Packet (SOP) and End Of Packet (EOP) markers (i.e. the *K.28.1* and *K.28.6* special 8b10b symbols, respectively) mark the boundaries of each packet. The EOP marker can be optionally dropped between back-to-back packets (see *cfg_eop_en_o* from Table 2.4). If the Busy-On/Off flow control mechanism is enabled (see *cfg_busy_en_o* from Table 2.4) other two special 8b10b symbols (i.e. *K.28.2* and *K.28.3* for Busy-On and Busy-Off, respectively) are sent between the L1 packets to signal the state of the associated VMM Capture FIFOs. If one of the occupancy levels rises above the *busy_on_limit_o* (see Table 2.4) threshold a Busy-On symbol is issued. Once this happened, when all the occupancy levels fall below the *busy_off_limit_o* (see Table 2.4) threshold a Busy-Off symbol is issued and so on. Similar to VMM3, the ROC's data are transmitted most significant symbol first. The formats of the ROC input and output packets are detailed in Section 2.3.2.

The TTC stream, depicted in Figure 2.11 as waveforms, is received on a single e-link operating at 320 Mbps as uncoded (**not** 8b10b encoded) data organized into bytes. Similarly, the bits are tagged with one letter indicating the position within the byte and one digit as the byte number. Bit a is the MSB while bit h is the LSB. The positive edge of the internal BC (i.e. 40 MHz) clock signal determines the byte alignment within the stream. Thus, in every BC clock cycle a new TTC byte is formed. The *ttc_stream_i* is supplied to the TTC deserializer directly from the ROC's input differential pad while the pacing internal BC and RO (i.e. 160 MHz) clock signals are supplied by the ePLL (see Figure 2.2). This ePLL introduces an unknown phase shift relative to the GBTx-supplied BC clock. Nevertheless, the internal ROC clock signals have independently configurable phases.

It was and still is not clear from the GBTx manual [35] which bits of the 8-bit frame defined by the 40 MHz e-link BC clock belong to which BC. The confusion started from Figure 2.12 from this document where it is uncertain if bits '1' and '0' from e-Link data 320 Mb/s are from the current BC (delimited by the e-Link clock 40 MHz) or the previous one. By the same logic, it is also uncertain if this behavior extends to the other bits.

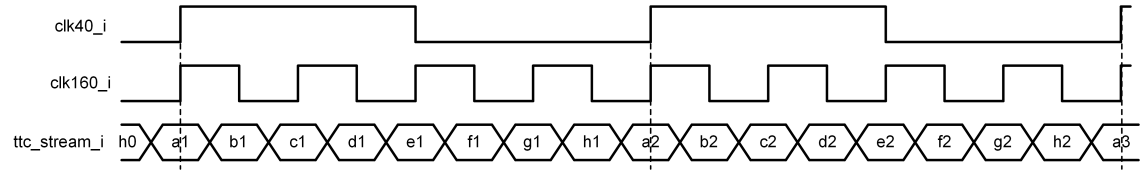


Figure 2.11: Waveforms depicting the TTC stream format.

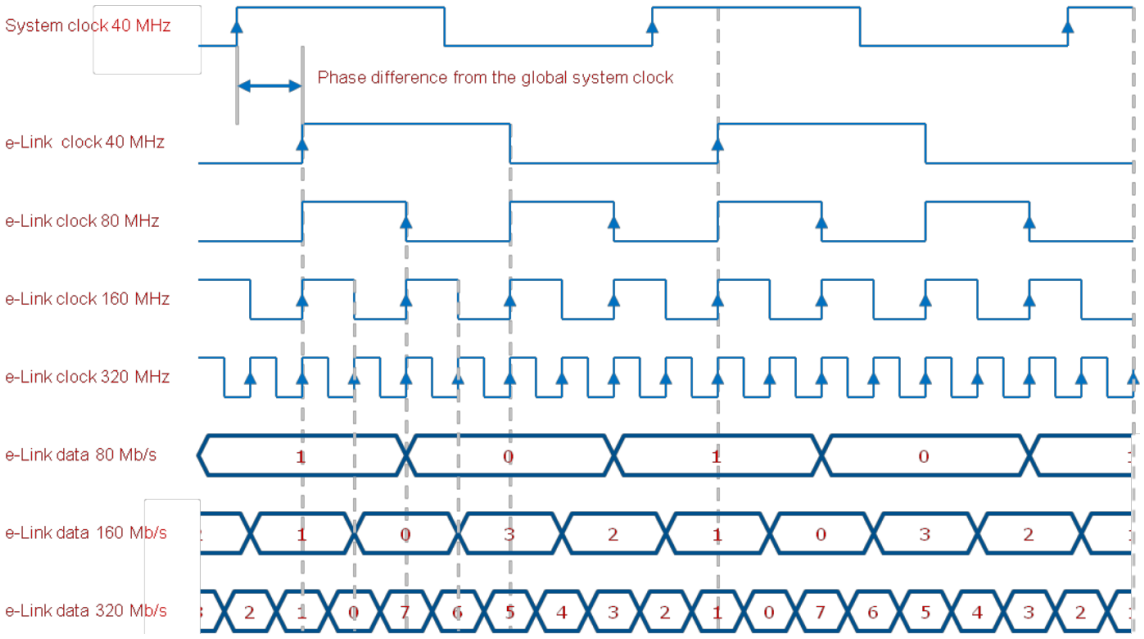


Figure 2.12: Waveforms depicting GBTx output e-link clock and data signals [35].

Thus, the TTC deserializer has four modes (alignment and bit order) of forming TTC bytes, depicted in Table 2.5. The x value represents the symbol from the previous BC clock cycle. The resulted TTC byte is next interpreted as depicted in Table 2.6. The four modes translate to the four scenarios from Figure 2.13. The internal BCID counter is depicted as a timing reference, considering that *bc_rollover_o* (see Table 2.4) is 3563. The resulted TTC byte (i.e. *ttc_word*) is paced by the internal RO clock signal but it always maintains its value for four cycles. Its bits representing TTC commands can thus be safely sampled in the BC clock domain.

ttc_starting_bits_i	**TTC byte**
0b00	cx, dx, ex, fx, gx, hx, ax, bx
0b01	cx, dx, ex, fx, g(x-1), h(x-1), ax, bx
0b10	cx, dx, e(x-1), f(x-1), g(x-1), h(x-1), ax, bx
0b11	cx, dx, ex, fx, gx, hx, a(x+1), b(x+1)

Table 2.5: The four modes in which the TTC deserializer forms the TTC bytes using the same notation convention as in Figure 2.11.

The mode is selected through *ttc_starting_bits_o* from Table 2.4. In Figure 2.12 if bits '1' and '0' are from the current BC then the mode should be set to 0b00. If they belong to the previous BCthen the correct option should be 0b11. Mode 0b01

7 (MSB)	6	5	4	3	2	1	0 (LSB)
L1A	SR	TP	ECR	BCR/OCR	L0A	SCA reset	EC0R

Table 2.6: The interpretation of the TTC byte supplied by the TTC deserializer in one of the four arrangements detailed in Table 2.5.

assumes bits '7', '6', '5', '4', '1' and '0' are from the current BC but 'bits '3' and '2' correspond to the next BC. 0b10 assumes bits '7', '6', '1' and '0' are from the current BC while bits '5', '4', '3' and '2' are from the next BC. In all four cases, the order of bits that form the TTC word is '7', '6', '5', '4', '3', '2', '1', '0'. What varies is from which BC (current or previous) bits '5', '4', '3', '2', '1' and '0' are from. Bits '7' and '6' are always from the current BC.

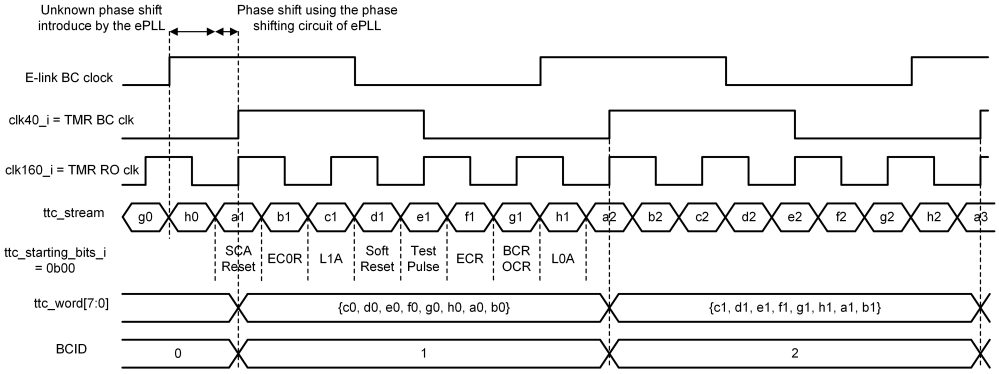
During the design phase for the ROC internal logic, one of the objectives was for it to have higher data throughput than its interfaces. The output interfaces should represent the bottleneck. Since the input data channels are twice as many as the output ones at the same speed, the ROC cannot cope with the saturation of all the input interfaces. As long as the SROC channels can saturate the output interfaces the objective is met.

2.3.2 ROC data formats

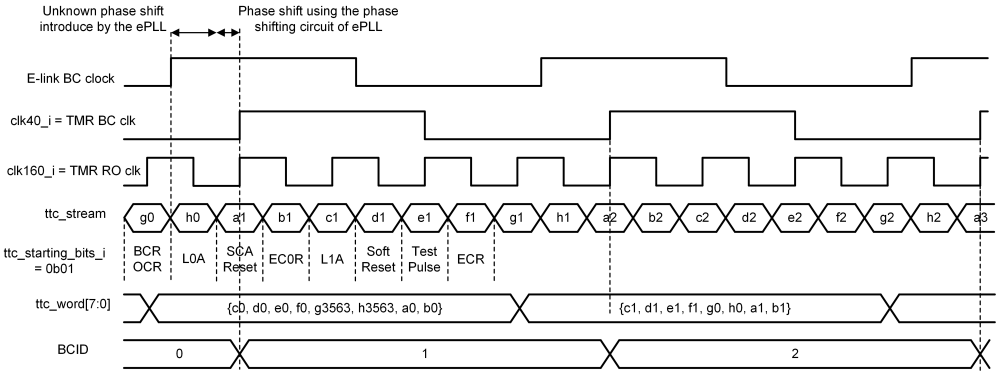
This section presents the formats of the input and output packets, explains their fields and how they are buffered inside the ROC. Some of them are summarized in [79], while [174] includes a complete presentation. Nevertheless, they are essential for understanding the ROC functional features, its mathematical model for operating in steady-state, its behavior in an environment with neutron irradiation and the chosen sizes for its FIFOs.

There are two types of ROC input packets, depending on the presence of VMM3 channel data: L0 null-events which do not contain any such data and L0 hit packets which contain data from at least one channel. Both types start with a 16-bit header word representing the L0 trigger information that caused the formation and transmission of the packet in the source VMM3 chip. An L0 null-event contains only the header word. It signals to the ROC that the source VMM3 received the L0 trigger pulse but it did not have any associated detector data rather than appearing inactive. In an L0 hit packet, the header is followed by at least one 32-bit hit word. Since one VMM3 chip is responsible for at most 64 detector channels and each hit word corresponds to one channel, the maximum number of hit words in an input packet is 64. The uncoded buffering format of the input packet words is detailed in Figure 2.14. Both types of words occupy one address in the VMM Capture FIFO buffer and are padded by the receive logic (i.e. the VMM Capture *Assembler* module) with an MSB that signals the end of the L0 packet. The 16-bit header is shifted to the left with 16 positions, the resulted 16 LSBs being all zero.

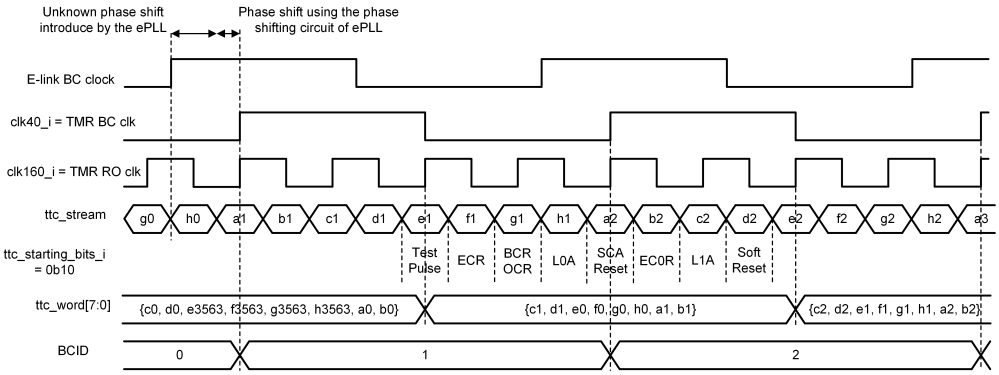
The header word contains the entire 12-bit BCID value and the two LSBs of the OrbitID counter corresponding to the BC selected by the L0A. The downstream TDAQ system can infer the full 32-bit wide OrbitID counter value. The parity bit P covers these two fields. The received L0 headers with incorrect parity are nevertheless buffered into the FIFO. The VMM Capture logic checks the parity of



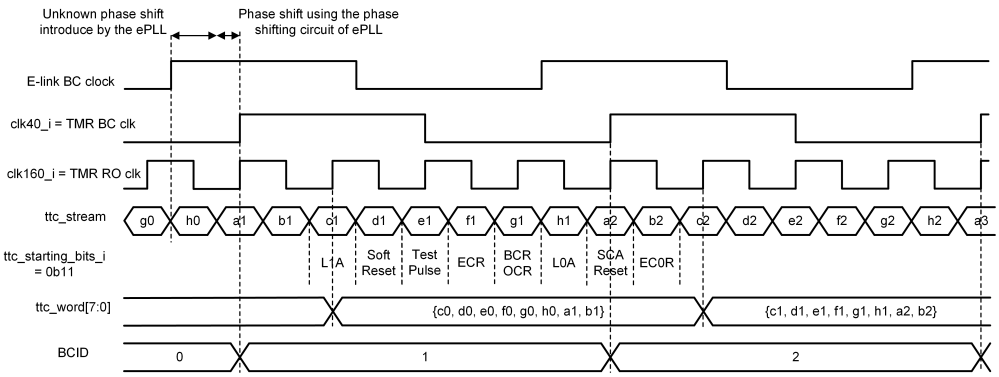
(a) *ttc_starting_bits_o* = 0b00



(b) *ttc_starting_bits_o* = 0b01



(c) *ttc_starting_bits_o* = 0b10



(d) *ttc_starting_bits_o* = 0b11

Figure 2.13: The resulting four modes of interpreting the TTC stream, based upon the information from Figure 2.11 and Tables 2.5 and 2.6.

	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
header	EOP	V	P	orbit(2)		BCID(12)												16'b0															
hit data	EOP	1	P	0	T	CHAN(6)						ADC (10)												TDC (8)						N	rel BCID(3)		

Figure 2.14: The buffering format of the input data packets [174].

the headers and if the parity is incorrect, the corresponding parity error counter from the Config module is incremented (see *incr_parity_vmm_x_i* from Table 2.4). When the SROC's Packet Builder FSM reads a header word it will recheck its parity. The entire packet is dropped if the parity is wrong. Thus, a subsequent error in the header word (e.g. from an SEU in the SRAM) will not cause the loss of synchronization between the L1 triggers and the L0 data, with the drawback of increased processing time and loss of the data of the affected packet. The V bit signals that the VMM3 has no detector data for the corresponding BC because its channel FIFOs overflowed. Thus, only the L0 null-events can have the V bit set.

The channel that produced a hit word is specified in its 6-bit *CHAN* field. The T bit (i.e. truncate) can only be set in the last hit word of an L0 packet and signals the truncation of the packet when the configurable maximum number of hit words was reached within VMM3. The 10-bit *ADC* value represents the digitized peak value of the voltage signal produced by the detector, whereas the 8-bit *TDC* value represents the time value when the detector voltage signal is above the configurable threshold. These two values are used to estimate the charge produced within the detector by the passing of a muon. The N bit (i.e. neighbor) indicates that the hit word is present because the detector signal of the adjacent channel was above the threshold. The L0 data selection within the VMM3 happens by finding detector data within BCID windows of configurable size (with a maximum size of 8 BCs) around the triggered BC. Therefore, the 3-bit *relative BCID* hit word field indicates the actual BC of the hit relative to the BC information from the header. The hit word parity bit is computed over *ADC*, *TDC*, N and the BCID header value added to the *relative BCID*. It does not cover the *CHAN* field.

When the L1A command is asserted, the TTC Capture module writes the current two LSBs of the OrbitID counter concatenated with the 12-bit BCID value and with the 16-bit L1ID counter value in the TTC FIFOs (as long they are not full). This format is depicted in Figure 2.15 and represents the L1 trigger information that will be processed by the SROCs. There are no parity or checksum fields. The 16-bit L1ID counter value represents the least significant subset of the 38-bit L1ID. Similar to the OrbitID, the full value is inferred by the downstream TDAQ system.

29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
orbit (2)		BCID (12)												L1ID (16)															

Figure 2.15: The buffering format of the L1 triggers within the TTC FIFOs [174].

The format for buffering the L1 packets in the SROC FIFO is depicted in Figure 2.16, before the insertion of the special protocol symbols (i.e. SOP, EOP, Busy-On, Busy-Off and *comma*) and the 8b10b encoding. There are eight types of words, each occupying one address in the Packet/SROC FIFO and being padded with an auxiliary, not to be sent MSB that signals the end of the packet: a 16-bit null-event, two types of 32-bit hit headers, four types of hits (two 32-bit wide and the other two 24-bit wide) and one 32-bit trailer.

From the point of view of the contained data, there are two types of hit words:

	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
null evnt	1	0	1	ROC ID (6)						LEVEL-1 ID (8)								16'b0															
hit header	0	0	0	orbit(2)			BCID (12)											LEVEL-1 ID (16)															
hit header no TDC	0	1	0	orbit(2)			BCID (12)											LEVEL-1 ID (16)															
hit data	0	P	N	rel BCID(3)			VMMid(3)			CHAN (6)						ADC (10)										TDC (8)							
hit data no TDC	0	P	N	rel BCID(3)			VMMid(3)			CHAN (6)						ADC (10)										8'b0							
dummy hit data	0	P	0	3'b0			VMMid(3)			6'b0						ADC (10) = 0x3ff										8'b0							
trailer	1	0	TO	VMM missing data flags (8)								L0 ID (4)				length (no. hits) (10)										checksum (8)							

Figure 2.16: The buffering format of the L1 packets within the SROC FIFOs [174].

normal hits which are based upon the received L0 data (i.e. they contain actual input data) and dummy hits which are generated by the SROC to signal specific overflow values within the associated VMM3 ASICs. The dummy hits contain specific field values to differentiate them from the normal hits words: all the fields and bits are set to 0 except the *ADC* which has all its 10 bits asserted, the *VMMid* which indicates the source VMM3 and the parity bit. The *ADC* field of a normal hit word is saturated inside the SROC to 0x3FE. From the point of view of the size, there are two types of hit words: 24 and 32-bit wide depending on the presence of the 8-bit *TDC* field. This field can be dropped in the output hit words to conserve bandwidth. This is useful in an sTGC system where it contains no essential information. The total four hit word types result from combining the content source and the size criteria. Figure 2.16 depicts only three of the hit word types, but the 24-bit wide dummy hit word has always the same buffering format as the 32-bit wide one. The difference is that the SROC will transmit only the 3 most significant bytes from the first one as opposed to all 4 bytes from the second one. In the buffering format from Figure 2.16, the 24-bit hit words are left-shifted with 8 positions and the resulting 8 LSBs are all 0.

One packet cannot contain both 24 and 32-bit wide hit words. Thus, based upon the size of the contained hit words, the two types of hit headers are specified. They differ only by one bit (i.e. bit number 31). The 16-bit *Level-1 ID* field represents the trigger count value and is copied from the TTC FIFO. All output hit word types contain a new 3-bit field named *CHAN* which indicates the source VMM Capture channel. It replaces *R*, T and the set bit (i.e. position 31) from the input hit word format. The flags are instead reported in the 32-bit trailer word. All the L1 packets that are not null-events end with a trailer. The trailer contains a 1-bit timeout flag (i.e. the *TO* bit), an 8-bit checksum, a 10-bit field that specifies the total number of hit words (i.e. the *length* field), a 4-bit wide *L0ID* field that is always zero and an 8-bit *VMM missing* field in which each VMM Capture module has a 1-bit flag that is raised whenever data is missing for the current trigger. Bit *E* is always 0 because it was a reserved bit for a possible format extension that never happened.

The intended role for the *L0 ID* field (the four LSBs of L0ID) was to indicate the correspondence between the L0 and L1 triggers (identical if only one level is used). To conserve bandwidth, the VMM3 does not include this value within its header. Thus, additional logic and a FIFO is required in the TTC Capture module to determine the value. Every time an L0A command is decoded, an L0A counter is incremented and its previous value is pushed alongside the corresponding BC information (i.e. the 16-bit BCID and 2-bit OrbitID values) in an additional L0 FIFO. This counter is cleared when the EC0R TTC command is asserted. The latency of the L1 trigger relative to the L0 trigger is an additional configuration value within the Config module. The TTC Capture logic is keeping the occupancy level of the additional L0 FIFO equal to the corresponding relative trigger latency

value expressed in BCs. Each time an L1 command is received, the four LSBs of the oldest L0ID value are also added in the four TTC FIFOs (i.e. concatenated to the format from Figure 2.15). This was considered too complex for a little gain because the downstream TDAQ system can infer the correspondence between the two trigger levels. Thus the 4-bit trailer *L0 ID* field remains unused.

The possible L1 packet word compositions are: (i) a single 16-bit L1 *null-event*; (ii) a 32-bit *hit header* or *hit header no TDC* word followed by a 32-bit *trailer* word that signals data loss; (iii) a 32-bit *hit header* or *hit header no TDC* word followed by at least one *hit* word (*dummy* or not) of corresponding size and a 32-bit *trailer* word at the end. The 16-bit L1 null-event is formed when the L1 trigger matches L0 null-events with cleared *V* bits in all the associated VMM Capture FIFOs. The L1 null-event has two fields: the 8-bit *Level-1 ID* field which represents the LSBs of the L1ID trigger count value (copied from the TTC FIFO) and the 6-bit *ROC ID* field which represents the configurable ROC identifier. The buffering and transmission of the L1 null-events packets are optional. In the buffering format from Figure 2.16, the L1 null-event is shifted to the left with 16 positions and the resulting 16 LSBs are cleared. Case (ii) shows that it is possible to have a packet that starts with a hit header that is immediately followed by a trailer word, without any hit words. In this case, the trailer's *length* field is zero and at least one of the *VMM missing* flags corresponding to the associated VMM Capture channels is set, indicating a lack of hit data due to an overflow or data corruption. E.g. if any of the L0 null-events that matched an L1 trigger has the *V* bit set then, instead of an L1 null-event, the resulting output packet consists of a header word plus a trailer in which the *VMM missing* field will signal the associated VMM3 ASICs with overflowed channel FIFOs.

There can be only one dummy hit word generated in the name of a certain VMM Capture channel in any L1 packet. Furthermore, there cannot be any normal hit words from that VMM3 in that packet. There can be normal hit words from the other VMM Capture channels for which no dummy hit words are generated. Thus, the maximum number of dummy hit words that an L1 packet can have is equal to the number of associated VMM Capture channels. Assuming that there is no need to generate dummy hits, the maximum number of hits that an L1 packet can have equals the number of associated VMM Capture modules times 64 (i.e. the maximum number of hit words from an L0 packet). The SROC's Packet builder FSM iterates through the associated VMM Capture channels one by one starting with the channel with the smallest index. After it finishes with a channel, it jumps to the next one, until the last channel is reached. Then, for the next L1 trigger the looping of the input channels starts all over again. Thus, the *VMMid* fields from the normal hit words of an L1 packet will always be in ascending order. The dummy hit words are always inserted after the normal ones. They are also transmitted in ascending order from the point of view of the *VMMid* field.

Each SROC includes a watchdog timer. When the *timeout* mechanism (i.e. the automatic disabling of the VMM Capture channels) is disabled, if the selected VMM Capture FIFO signals to the SROC that it is empty for an uninterrupted period that reaches the configurable *timeout* threshold, the SROC abandons the channel and jumps to the next associated VMM Capture. In this way, the SROC does not become stuck waiting for data on a malfunctioning channel. The corresponding packet trailer will have the *TO* bit set and the affected channel's corresponding *VMM missing* flag will also be set. The problematic channel is not disabled and it

will be probed at the next L1 trigger. When the *timeout* mechanism is enabled, the problematic channel will be automatically disabled. While in this state, the SROC will ignore the channel and the *TO* bit will be set in conjunction with an asserted *VMM missing* flag in all the output packets.

If the BCID value of an L0 packet has a specific value outside the normal interval [0, 3563] (i.e. 4072 called *MAGIC BCID* or *PATTERN*) in conjunction with an incorrect header parity then the VMM3 signals that its L0 trigger FIFO overflowed which means that it will not respond to a certain number of following L0 triggers. In this case, the ROC should not wait for data from that VMM3 and thus should not waste processing time. Furthermore, it should not risk reaching the *timeout* threshold and the channel should not be automatically disabled. Instead, the SROC will signal this situation by generating a dummy hit word in the name of the affected channel. The corresponding *VMM missing* flag is set. This behavior persists for each L1 trigger until the SROC finds an older or matching L0 header (i.e. in terms of the BC information).

A *VMM missing* flag is set for a VMM Capture in the following situations:

1. VMM3 signals an overflow in its L0 trigger FIFO by issuing a header with the special BCID value of 4072 or it does not respond to the L0 triggers due to a previous trigger FIFO overflow. This situation distinguishes in the resulting output packets by the presence of a dummy hit word for the affected VMM3 for as long as the situation persists.
2. VMM3 signals that its channel FIFOs overflowed through the *V* bit. In this case, there will be no hit words of any kind for the affected VMM3.
3. VMM3 signals the truncation of the L0 packet by a set *T* bit in its last hit word. The resulting output packet is characterized by the presence of at least one normal hit word from the affected VMM3.
4. VMM3 stopped sending packets and the receiving VMM Capture FIFO was empty for at least the timeout threshold time. There are no hit words from the affected VMM3 and the *TO* bit is asserted. If the timeout feature is enabled the corresponding VMM Capture channel is automatically disabled. In this case, until the channel is explicitly re-enabled, for each L1 packet the *TO* bit and the corresponding *VMM missing* bit will be set.
5. There isn't a corresponding L0 header to the searched L1 trigger. There will be no hit words of any kind for the affected VMM3. In the output packet, this situation is indistinguishable from the second and the next one. In normal operation, there should not be L1 triggers that do not match the issued L0 triggers. However, overflows might occur causing the first situation. Also, the L1 trigger data might be altered while being buffered within the TTC FIFO (e.g. due to SEUs). The SROC might search for an incorrect BC. Alternatively, the buffered L0 header might be corrupted (the next situation).
6. When the SROC reads a header from an associated VMM Capture FIFO it checks its parity. If the parity is incorrect, the entire packet is dropped. The corresponding output packet will be similar to the previous and the second situations. If the parity error occurred before the push within the VMM Capture FIFO, then the parity error counter for that channel was incremented.
7. When the VMM Capture FIFO has only one address free and a new hit word has been received, it will be buffered with its *T* bit set regardless of its initial value. The output is indistinguishable from situation 3.

The used parity is even but there is a configuration option that can change it to odd. The TTC FIFO data is not covered by any data integrity fields. There are no parity bits or checksum fields within an L1 null-event. The two types of output headers do not include any parity bits but the trailer's *checksum* field covers the entire data between SOP and EOP (or the next SOP). The 8-bit checksum is similar to the IPv4 header checksum [176] (i.e. the one's complement of the one's complement sum of the covered bytes). The parity bit of the normal hit word is adapted from the input, as follows: if the *TDC* field is kept and the received *ADC* field is different from 0x3FF then the parity bit is simply copied from the L0 hit word; if the *TDC* field is dropped or when the *ADC* field is saturated to 0x3FE, the output parity is recomputed to indicate the same correctness as when the hit was read from the VMM Capture FIFO. The correct dummy hit word indicates incorrect parity. If the checksum of an L1 packet is correct but a hit word indicates invalid parity then it means that one of the bits was incorrect when the hit word was read from the VMM Capture FIFO. If the checksum is also incorrect then the error happened when that packet was buffered within the SROC FIFO.

2.3.3 FIFO overflows, congestion and flow control

As shown in [42] the probability that data is lost in a queueing system with finite buffer equals the probability that the buffer is full. All ROC's FIFO full flags are reported within its Config module. A finite FIFO becomes full, regardless of its buffer space, if the average reading/popping/dequeuing rate is less than the average writing/pushing/enqueuing rate. The writing rate into the VMM Capture FIFO is determined by the received data. The writing rate into the TTC FIFO is also determined by an external factor: the L1 trigger rate. The reading rate of the SROC FIFO is determined by the configured transmission speed and the used protocol. The connecting element of all these FIFOs is the SROC's Packet Builder FSM. It determines the reading rate from the VMM Capture and TTC FIFOs and the writing rate into the SROC FIFO. The aim was to minimize data loss hence its processing rate was maximized.

In Section 2.4 the theoretical maximum SROC transmission rate is determined based upon the ROC's configuration, the presented interfaces, transmission protocols and data formats and the input traffic characteristics. This model is then validated in Chapter 3 where it is also proven that the SROC processing rate is higher than this maximum transmission rate.

The Packet Builder FSM implements the following congestion control mechanism: when the Packet FIFO signals that it is full, the FSM throttles (i.e. pushes data only when the FIFO is not full). This means that its processing rate decreases up to the transmission rate and as a result, the reading rates from the VMM Capture and the TTC FIFOs also decrease. If their writing rates are high enough, they will fill and data loss will occur. Thus a flow control mechanism was implemented in the SROC Streamer FSM. Depending on the fill levels of the associated VMM Capture FIFOs and the configured hysteresis thresholds, the special Busy-On/Busy-Off symbols are injected in the output data stream, if enabled. The idea is that when the trigger path receives the indication that in some ROIs the buffers are about to fill (via the Busy-On), it will reduce the L0A rates (and as a result the L1A rates) for those regions. This means that less L0 data and L1 triggers would be received.

The writing rates for the VMM Capture and TTC FIFOs are decreased. When all fill levels go below the lower threshold the issued Busy-Off would indicate that the normal triggers rates can be used.

The Packet Builder forwards the VMM3's congestion flags, i.e. the handling of the input V and T bits, the insertion of dummy hit words when the special header *PATTERN* is received and the *VMM missing* trailer flags. Additionally, the VMM Capture Assembler module uses the almost full signal from the FIFO, asserted when only one address is free, to truncate the current L0 packet. If a hit word is to be written then its T bit is set regardless of its received value.

2.3.4 ROC FIFOs dimensioning

The optimal depths for the ROC FIFOs were determined considering the worst-case parameters of the input traffic, the ROC's requirements and the available silicon die area. The aim was to minimize or even remove any possibility of data loss within the ROC while also using buffers as small as possible. Smaller buffers translate into a design that occupies less silicon die area and draws less power. In addition, more chips are being fitted on the same silicon wafer and a higher manufacturing yield is obtained (i.e. a fabrication defect in conjunction with a large design means that more wafer area is declared *bad* and becomes unusable).

Thus, the worst-case occupancy levels for all the ROC's FIFO categories (i.e. VMM Capture, TTC and SROC) were determined. An increased VMM3 to ROC effective throughput (i.e. the throughput of the actual packet data to be buffered) translates into more data being pushed into the corresponding VMM Capture FIFO. The more data is being buffered within these FIFOs the longer the associated SROC will spend processing the buffered L1 triggers. Also, more resulting L1 data would be pushed into the SROC's Packet FIFO and the effective SROC output effective throughput will increase.

The maximum VMM3 to ROC bandwidth utilization is reached when L0 packets of maximum size are transmitted as close as possible one after another, i.e. back-to-back with the mandatory two *commas* between them. The rate of *commas* is minimized. The largest VMM3 packet contains a header and 64 hit words and thus occupies 65 addresses in the VMM Capture FIFO (see Section 2.4.2). Considering the mathematical model from Section 2.4.3, the maximum rate of transmission of such L0 packets is 246.15 kHz. This model was determined strictly from the point of view of the transmission interface to the ROC, the data formats and the associated protocol. The assumption is that there is no limiting bottleneck or FIFO overflow within the VMM3, even though the VMM3-ROC joint requirements document [123] states that each VMM3 digital channel has a buffer depth of only 84 addresses, each one buffering data for one hit word.

The largest L1 trigger latency stated within the VMM3-ROC joint requirements document [123] and the ATLAS Front-End Interface Requirements document [177] is 60 μs. The resulting maximum occupancy level for the VMM Capture FIFO is $246.15 \times 10^3 \times 65 \times 60 \times 10^{-6} = 960$ addresses. To not cause data loss in this extreme case, a minimum buffer depth of $2^{10} + 1$ addresses must be used (2^{10} addresses are within the SRAM and the remaining one is register-based). The SROC does not process the packets from all its associated VMM Capture FIFOs simultaneously so the occupancy level will reach even higher values. Therefore, a buffer space of

$(2^{11}+1) \times 33$ bits from which $2^{11} \times 33$ bits are SRAM was chosen for each VMM Capture FIFO.

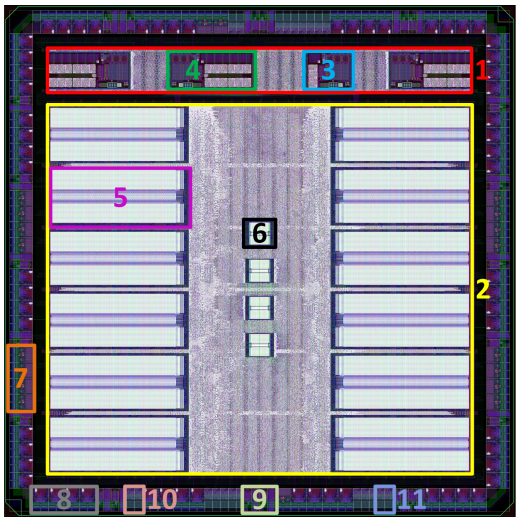
The scenario with the worst-case L1 trigger (and L0 packet) burst in a loop is detailed in Chapter 3, Algorithm 1. It was considered when sizing the TTC and SROC Packet FIFOs. One burst consists of 128 L1 triggers (and L0 packets) being issued almost back-to-back in 8 μs, followed by a pause of 120 μs. It results in a 1 MHz average rate but the SROC cannot process the triggers and packets as fast as they are enqueued into the FIFOs, as shown in Chapter 3. Consequently, the occupancy levels of the TTC and VMM Capture FIFOs rise. Concurrently, the resulting output packets are being written into the SROC Packet FIFO faster than they are transmitted, regardless of the interface speed. The output interface and the associated protocol are the bottlenecks within the ROC. The mathematical model predicts that in these conditions the ROC will not lose data as long as the constant number of hit words from the L0 packets is less than or equal to 6. It is demonstrated both mathematically and empirically that at 7 hit words in every L0 packet, the output bandwidth will be saturated, the ROC FIFOs will overflow and data loss will occur. If the number of L0 hit words is limited to 6 or below then no data is discarded. The occupancy levels of the TTC and SROC Packet FIFOs reach a maximum of 114 and 665 addresses, respectively (see Chapter 3). This was also observed in RTL and gate-level simulations during the design and implementation. Therefore, for the TTC FIFO, a buffer size of $(2^7+1) \times 30$ bits (from which $2^7 \times 30$ bits are SRAM) was chosen. For the SROC Packet FIFO, the same buffer used for the VMM Capture FIFO was employed. In this case, the depth is twice as big as needed but it simplified the layout floorplanning, as shown in Section 2.3.5. As with the VMM Capture FIFO, the SROC Packet FIFO buffer is larger than double the maximum expected fill level.

2.3.5 ROC layout and package

The layout of the ROC silicon die is depicted in Figure 2.17, with highlighted main regions and macros. The die is a square with an area of approx. 22.5 mm^2 (i.e. the side length is 4.744 mm) and has only one row of pads. The ROC has 232 pads from which 187 are used for IO. The layout is core limited slightly, meaning that the size of the core determines the die size and not the pad number. The total width of the pad filler cells is less than the narrowest IO or power pad.

Similar to its architecture, the core is divided into two areas based on their function: the *analog* and the *digital* parts. The digital part has an area of 13.3 mm^2 from which 64 % represents SRAM (i.e. approx. 8.5 mm^2). As described in Section 2.3.4, 12 identical buffers are used for the eight VMM Capture and four SROC Packet FIFOs: a 0.689 mm^2 dual-port, dual clock domain SRAM containing $2^{11} \times 33$ bits. Four dual-port, single clock domain $2^7 \times 30$ bits SRAMs represent the buffers of the four TTC FIFOs. Each has an area of 0.054 mm^2. The rest of the packet processing logic is contained in the remaining area of 4.8 mm^2 and consists of 83,079 combinational gates and 18,458 flip-flops. In Figure 2.17 the densities of sequential elements (i.e. flip-flops and latches) within the entire ROC can be visualized since they are highlighted with white border.

Initially, the silicon die was wire-bonded on a prototype testing Printed Circuit Board (PCB) as depicted in Figure 2.18a. An intermediate Quad Flat Package

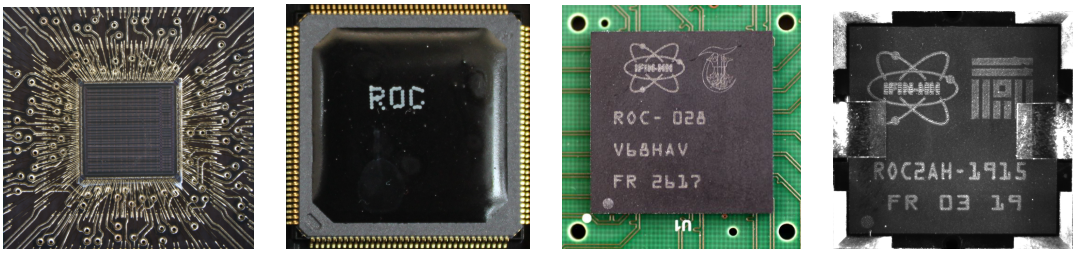


1. The *analog* part
2. The *digital* part
3. The ePLL for the internal clocks
4. One ePLL for 4 BC and 4 RO output clocks
5. One of the 12 dual-port, dual clock domain $2^{11} \times 33$ bits SRAMs used for the VMM Capture and SROC Packet FIFOs
6. One of the four dual-port, single clock domain $2^7 \times 30$ bits SRAMs used for the TTC FIFOs
7. Several input SLVS pads
8. Several output SLVS pads
9. Several power supply pads
10. Two output single-ended pads (the right one is the SEU pad)
11. Two input single-ended pads

Figure 2.17: ROC's layout with highlighted sequential cells and macroblocks [79].

(QFP) with 144 pins, illustrated in Figure 2.18b, was used for partial design validation. The final ROC package, depicted in Figures 2.18c and 2.18d, is a 16×16 Ball-Grid Array (BGA) package with a pitch of 1 mm and a ball diameter of 0.6 mm, resulting in a total footprint area of 289 mm^2.

For the ROC's high-speed SLVS IO interfaces, the receiver and transmitter e-link IP (Intellectual Property) blocks from [62] were used. For the other, slower interfaces, the standard library pads were used. In Appendix A.3 all the ROC IO interfaces are detailed.



(a) Wire-bonded (b) QFP (c) BGA on PCB (d) BGA in socket

Figure 2.18: A ROC die wire bonded on a prototype testing PCB (a), the intermediate QFP 144 package (b) and the final 16×16 BGA package soldered on a testing PCB (c) and in an open-top socket (d).

2.4 ROC steady-state model

2.4.1 ROC queueing theory model

Queueing theory represents the mathematical study of waiting lines or queues [42]. In a queueing system customers from a population arrive to receive some service. The service is provided by one or more servers. If no server is free, the customers join the waiting queue. When a server becomes free the next client receives service. The

terms *customer*, *server* and *queue* are generic. Within digital systems the *customers* can be IO requests, processes requiring CPU time, data packets to be transmitted and/or processed, etc. The *queue* does not indicate necessarily a First-In-First-Out queuing discipline. E.g. the next customer can be selected randomly, after a priority schema or after a Last-In-First-Out strategy (stack). Queueing theory describes features like average waiting time, server utilization, conditions for a steady-state system, optimal queue depth, scheduling strategies, etc.

The considered queueing theory model for the ROC is depicted in Figure 2.19. While the ASIC has four identical SROC modules (i.e. the number of identical processing and transmission servers is four, $c = 4$ in Kendall notation [136]), only one is shown, the others being identical. The VMM Capture FIFOs associated with the SROC are numbered from 1 to m. Each SROC contains two servers: the Packet Builder FSM that constructs the L1 events based upon the L0 events from the associated FIFOs and the received L1 triggers (i.e. the processing server) and the Streamer FSM that implements the sending protocol for the completed output events (i.e. the transmission server).

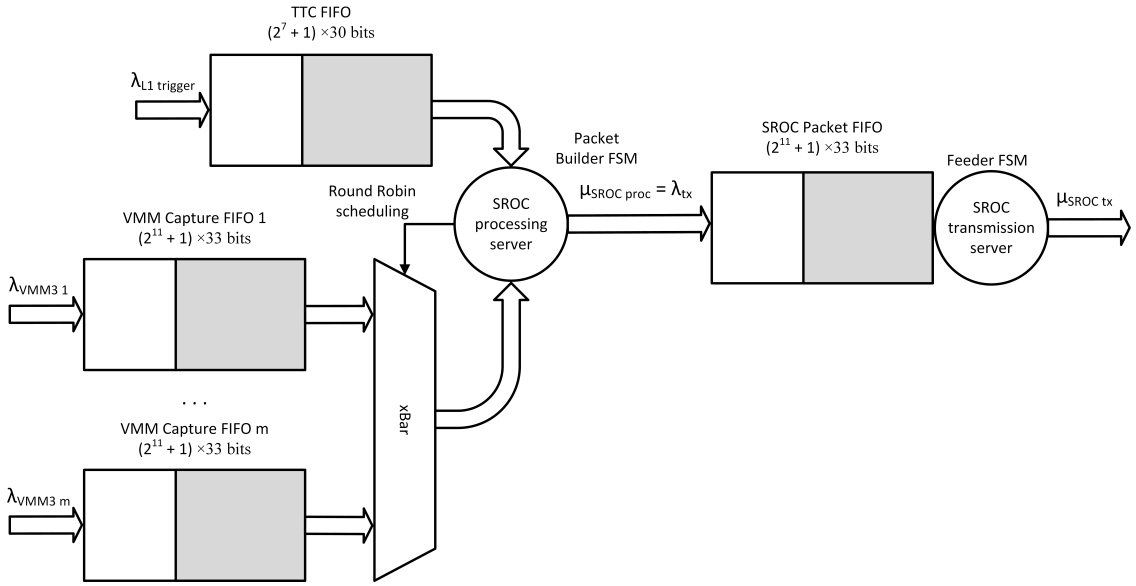


Figure 2.19: Queueing theory model for the ROC (one SROC module is depicted).

Since all the VMM3 ASICs associated with one ROC respond to the same L0 trigger commands and assuming that no overflows happen in these chips, causing the drop of L0 triggers, the average input data rates are equal: $\lambda_{\text{VMM3 1}} = ... = \lambda_{\text{VMM3 i}} = ... = \lambda_{\text{VMM3 m}} = \lambda_{\text{VMM3}} = \lambda_{\text{L0 trigger}}$, $\forall i \in \mathbb{N}, 1 \leqslant i \leqslant m$.

When only one trigger level is deployed within the ATLAS TDAQ system, the average arrival rate of L1 triggers is the same: $\lambda_{\text{L1 trigger}} = \lambda_{\text{L0 trigger}} = \lambda_{\text{VMM3}}$. When two stages are deployed for the hardware trigger, the ROC is responsible for the second level matching within the NSW readout system. Thus, the condition $\lambda_{\text{L1 trigger}} \leqslant \lambda_{\text{VMM3}} = \lambda_{\text{L0 trigger}}$ covers both HL-LHC upgrade phases. The selection ratio is defined as $sel = \frac{\lambda_{\text{L1 trigger}}}{\lambda_{\text{VMM3}}} = \frac{\lambda_{\text{L1 trigger}}}{\lambda_{\text{L0 trigger}}}$, $0 \leqslant sel \leqslant 1$ and has the requirement that $\lambda_{\text{VMM3}} = \lambda_{\text{L0 trigger}} > 0$.

The SROC Packet FIFO is the buffer between the two servers. The condition for it to not fill and consequently throttle the processing server is $\rho_{\text{SROC tx}} \leqslant 1$, where $\rho_{\text{SROC tx}}$ is the transmission server's utilization, defined as $\rho_{\text{SROC tx}} = \frac{\lambda_{\text{tx}}}{\mu_{\text{SROC tx}}}$.

$\mu_{\text{SROC tx}}$ is the average transmission rate when the server is busy while λ_{tx} is the average arrival rate of *customers* (i.e. the L1 packets from the processing server). But $\lambda_{\text{tx}} = \mu_{\text{SROC proc}}$, where $\mu_{\text{SROC proc}}$ is the average Packet Builder's service rate of L1 triggers (or the rate of enqueueing L1 packets into the SROC Packet FIFO).

Similarly, the TTC and VMM Capture FIFOs will not fill (and consequently data will not be lost) as long as the following condition is true: $\rho_{\text{SROC proc}} \leqslant 1$. The processing server utilization is defined as $\rho_{\text{SROC proc}} = \frac{\lambda_{\text{L1 trigger}}}{\mu_{\text{SROC proc}}}$. The conditions for not filing any ROC FIFOs translate to equation 2.1. For no data loss within the ROC, the average arrival rate of L1 trigger commands must be smaller than or at worst equal to the average rate the SROC can service these triggers, which in turn must be smaller than or at worst equal to the maximum rate of transmission of the resulting L1 events. Data loss does not occur within the SROC Packet FIFO if $\mu_{\text{SROC proc}} > \mu_{\text{SROC tx}}$ but as soon as the FIFO is full $\mu_{\text{SROC proc}}$ will be decreased to $\mu_{\text{SROC tx}}$.

$$\lambda_{\text{L1 trigger}} \leqslant \mu_{\text{SROC proc}} \leqslant \mu_{\text{SROC tx}} \tag{2.1}$$

The maximum average SROC transmission rate is dictated by the interface protocol and depends on the average size of the input events and the ROC's configuration. The maximum average SROC processing rate depends on the Packet Builder design, the state of the Packet FIFO (i.e.when full it throttles the server), the ROC configuration, the selection ratio *sel* and the average size of the L0 events. The maximum average L1 trigger arrival rate $\lambda_{\text{max L1 trigger}}$ is not dependent on ROC factors, except indirectly through the Busy-On/Off mechanism.

The maximum ROC performance (i.e. the maximum average rate of L1 triggers that does not cause any data loss within ROC) is achieved if the processing server is not the bottleneck in the chain from Figure 2.19. This means that the maximum average servicing rate for the L1 triggers must be higher than or equal to the maximum average transmission rate of the resulted events: $\mu_{\text{SROC\_max\_proc}} \geqslant \mu_{\text{SROC\_max\_tx}}$. Thus, the condition for no data loss from equation 2.1 in conjunction with the condition for maximum obtainable performance translates to:

$$\lambda_{\text{max L1 trigger}} \leqslant \mu_{\text{SROC\_max\_tx}} \leqslant \mu_{\text{SROC\_max\_proc}} \tag{2.2}$$

In equation 2.2 it was considered that there is no drawback for a maximum average processing rate strictly greater than the maximum average transmission rate since the SROC Packet FIFO becoming full does not cause data loss.

The SROC utilization as a whole can be defined as $\rho_{\text{SROC}} = \frac{\lambda_{\text{L1 trigger}}}{\mu_{\text{SROC tx}}}$ or as $\rho_{\text{SROC}} = \frac{m \cdot \lambda_{\text{VMM3}}}{\mu_{\text{SROC tx}}}$. The first variant considers that the actual clients are the L1 triggers. The SROC does not process the received and buffered L0 data without them. In this case, ρ_{SROC} has a similar meaning as in the classical queueing theory models. No data loss occurs while $\rho_{\text{SROC}} \leqslant 1$. The second definition strictly considers the input and output data packets rates. The L1 triggers are viewed as latency in the processing of the customers (i.e. the VMM3 packets). In this case, a value greater or equal to 1 does not necessarily indicate data loss.

In the following subsections, the average sizes for both the input and output packets are determined. Using these values, the maximum transmission rate is computed. Next, the maximum processing rate is deduced.

2.4.2 ROC packet sizes

The following equations express the size of a VMM3 packet in sent bytes to the ROC and occupied addresses and bytes within VMM Capture FIFO:

$$\begin{aligned} \text{size}_{VMM3\_pkt\_FIFO\_addr} &= n + 1 && \text{[FIFO addresses]} \\ \text{size}_{VMM3\_pkt\_FIFO\_bytes} &= 4 \cdot n + 2 && \text{[FIFO bytes]} \\ \text{size}_{VMM3\_pkt\_sent\_bytes} &= 4 \cdot n + 4 && \text{[sent bytes]} \\ n &= \#\text{ hit words}, & n \in \mathbb{N},\ 0 \leqslant n \leqslant 64 \end{aligned}$$

The VMM3 always outputs 2 commas after each packet, hence the two extra bytes considered for the transmission line.

The following equations express the size of an SROC packet in occupied addresses and bytes of the SROC Packet FIFO and the sent bytes on the transmission line to the L1DDC GBTx:

$$\begin{aligned} \text{size}_{SROC\_hdr\_pkt\_FIFO\_addr} &= n + 2 && \text{[FIFO addresses]} \\ \text{size}_{SROC\_null\_ev\_pkt\_FIFO\_addr} &= 1 && \text{[FIFO addresses]} \\ \text{size}_{SROC\_hdr\_pkt\_FIFO\_bytes} &= h \cdot n + 8 && \text{[FIFO bytes]} \\ \text{size}_{SROC\_null\_ev\_pkt\_FIFO\_bytes} &= 2 && \text{[FIFO bytes]} \\ \text{size}_{SROC\_hdr\_pkt\_sent\_bytes} &= h \cdot n + 10 && \text{[sent bytes]} \\ \text{size}_{SROC\_null\_ev\_pkt\_sent\_bytes} &= 4 && \text{[sent bytes]} \\ n &= \#\text{ of hit words}, & n \in \mathbb{N},\ 0 \leqslant n \leqslant 64 \cdot m \\ m &= \#\text{ of associated VMM3s}, & m \in \mathbb{N},\ 1 \leqslant m \leqslant 8 \\ h &= \#\text{ of bytes for a hit word}, & h \in \{3, 4\} \end{aligned}$$

For both the null event and header packets it was assumed that each one is preceded by an SOP and succeeded by an EOP, hence the two extra bytes considered for the transmission line.

2.4.3 VMM3 and SROC maximum transmission rates

To estimate the VMM3 and SROC maximum rates of packet transmission, some assumptions and considerations were made for simplification. One SROC has m associated VMM3 ASICs, each one having the independent probability p, $0 \leqslant p \leqslant 1$, that it will output an empty packet (i.e. with no hits, $n = 0$) in response to an L0 trigger. Furthermore, for each VMM3, the number of hit words within the non-empty packets is described by the independent discrete random variable n, $n \in \mathbb{N},\ 1 \leqslant n \leqslant 64$. Thus, the probability that the resulting L1 packet is a null-event equals the probability that all m VMM3s transmitted null-events:

$$\text{P}_{L1\_null\_ev\_pkt} = \text{P}_{all\_VMM3\_pkt\_empty} = p^m$$

Also, the probability that the resulting L1 packet has detector data (i.e. hit words) from $i, 1 \leqslant i \leqslant m$, of the associated VMM3 ASICs equals the probability that i out of m VMM3 ASICs responded with non-empty packets[4]:

[4]The $\binom{n}{k}$ notation represents the combination of n taken as k: $\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$.

$$\mathrm{P}_{hdr\_pkt\_i\_sources} = \mathrm{P}_{i\_VMM3\_pkt\_not\_empty} = \binom{m}{i} \cdot (1-p)^i \cdot p^{m-i}, \forall i \in \mathbb{N},\ 1 \leqslant i \leqslant m$$

If k represents the maximum number of consecutive packets that the SROC can transmit back to back, $1 \leqslant k \leqslant 255$, the maximum average SROC transfer rate is given by $\mu_{SROC\_max\_tx} = k \cdot r$, where r is the maximum average transmission rate of groups of k back-to-back packets and two *commas*. The sending of *Busy-On* and *Busy-Off* control symbols is not considered since it depends on the VMM Capture FIFO occupancies. To compute r, the average size of the group, in transmitted bytes, must be determined. This size is considered $\mathrm{size}_{avg\_L1\_pkt\_group} = k \cdot \mathrm{size}_{avg\_L1\_pkt} + (3-e)$, where e is 1 if the sending of EOP between back-to-back packets is enabled, 0 otherwise. Considering the previous formulas, the average size of the SROC packet, in transmitted bytes, is described by the following equation:

$$\mathrm{size}_{avg\_L1\_pkt} = p^m \cdot (3+e) \cdot o + \sum_{i=1}^{m} \binom{m}{i} \cdot (1-p)^i \cdot p^{m-i} \cdot (9+e+i \cdot \bar{n} \cdot h)$$

$o = 1$ if the sending of L1 null-events is enabled, 0 otherwise

$\bar{n}$ = mean of the random variable n

Thus, the average size of the group, in transmitted bytes, is:

$$\mathrm{size}_{avg\_L1\_pkt\_group} = k \cdot [\sum_{i=1}^{m} \binom{m}{i} \cdot (1-p)^i \cdot p^{m-i} \cdot (9+e+i \cdot \bar{n} \cdot h) + \\ + p^m \cdot (3+e) \cdot o] + 3 - e$$

The time for sending an average group of k back-to-back packets and two *commas*, in seconds, is:

$$\mathrm{t}_{tx\_avg\_L1\_pkt\_group} = \frac{\mathrm{size}_{avg\_L1\_pkt\_group}}{\frac{v}{10}} = \\ = \frac{10 \cdot [k \cdot \sum_{i=1}^{m} \binom{m}{i} \cdot (1-p)^i \cdot p^{m-i} \cdot (9+e+i \cdot \bar{n} \cdot h) + k \cdot p^m \cdot (3+e) \cdot o + 3 - e]}{v},$$

where v is the SROC throughput (8b10b encoded) in bps.

The maximum SROC transfer rate (in pkt/s) is achieved if the groups of k packets and two *commas* are sent back-to-back (i.e. no other *comma*, *Busy-On* or *Busy-Off* control symbols are sent), resulting in the following equation:

$$\mu_{SROC\_max\_tx} = \\ = \frac{k \cdot v}{10 \cdot [k \cdot \sum_{i=1}^{m} \binom{m}{i} \cdot (1-p)^i \cdot p^{m-i} \cdot (9+e+i \cdot \bar{n} \cdot h)\ + k \cdot p^m \cdot (3+e) \cdot o + 3 - e]}$$

The variables used within the formula are detailed in Table 2.7. The selection ratio *sel* does not appear in the formula because only the output packets are considered. However, to reach and maintain the maximum SROC transfer rate, *sel*

must be high enough so that the SROC Packet FIFO always contains at least one complete L1 packet. The insertion of *dummy* hit words in response to *PATTERN* special BCID values is also not considered.

Var.	Description
e	the configured enable state for the transmission of EOP symbols between the back-to-back L1 packets, $e \in \{0, 1\}$, by default $e = 1$ for each SROC.
h	the configured number of bytes of an L1 hit word (i.e. with or without the TDC field), $h \in \{3, 4\}$, by default $h = 4$ for each SROC.
k	the configured maximum threshold for the number of back-to-back L1 packets, $1 \leqslant k \leqslant 255$, by default $k = 255$.
m	the number of associated VMM Capture channels, $1 \leqslant m \leqslant 8$, by default $m = 2$.
$\bar{n}$	the mean of the discrete random variable $n \in \mathbb{N},\ 1 \leqslant n \leqslant 64$ which represents the number of L0 hit words in the non-empty VMM3 packets.
o	the configured enable state for the transmission of L1 null-event packets, $o \in \{0, 1\}$, by default $o = 1$ for each SROC.
p	the probability that a VMM3 would output an L0 null-event in response to an L0A.
v	the configured transmission bit rate for the SROC, $v \in \{80, 160, 320, 640\} \times 10^6$ bps, by default $v = 640 \times 10^6$ bps for each SROC.

Table 2.7: Description of the variables from the maximum SROC and VMM3 transmission rate formulas.

For determining the maximum VMM3 output rate it is assumed that the ASIC can saturate its output interface (i.e. transmit the packets back-to-back). The average L0 packet size, in transmitted bytes, is:

$$\text{size}_{avg\_L0\_pkt} = 4 \cdot (1 - p) \cdot (\bar{n} + 1) + 4 \cdot p = 4 \cdot [(1 - p) \cdot (\bar{n} + 1) + p]$$

Thus, the maximum VMM3 output rate, in pkt/s, is:

$$\lambda_{VMM3\_max\_tx} = \frac{v_{VMM3}}{40 \cdot [(1 - p) \cdot (\bar{n} + 1) + p]}$$

Since $v_{VMM3} = 640 \times 10^6$ bps, the equation becomes:

$$\lambda_{VMM3\_max\_tx} = \frac{1.6 \cdot 10^7}{(1 - p) \cdot (\bar{n} + 1) + p}$$

In Figure 2.20 the theoretical VMM3 and SROC maximum transmission packet rates are plotted as functions of $\bar{n}$. In this case $p = 20\%$. The continuous black trace corresponds to the default ROC configuration. One can observe that even if only a single VMM3 is associated with an SROC channel, the overhead of the output format and the transmission protocol are the limiting factors. In Chapter 3, Section 3.2.4 these conclusions and the maximum rate formulas are validated on the real-world ROC ASIC.

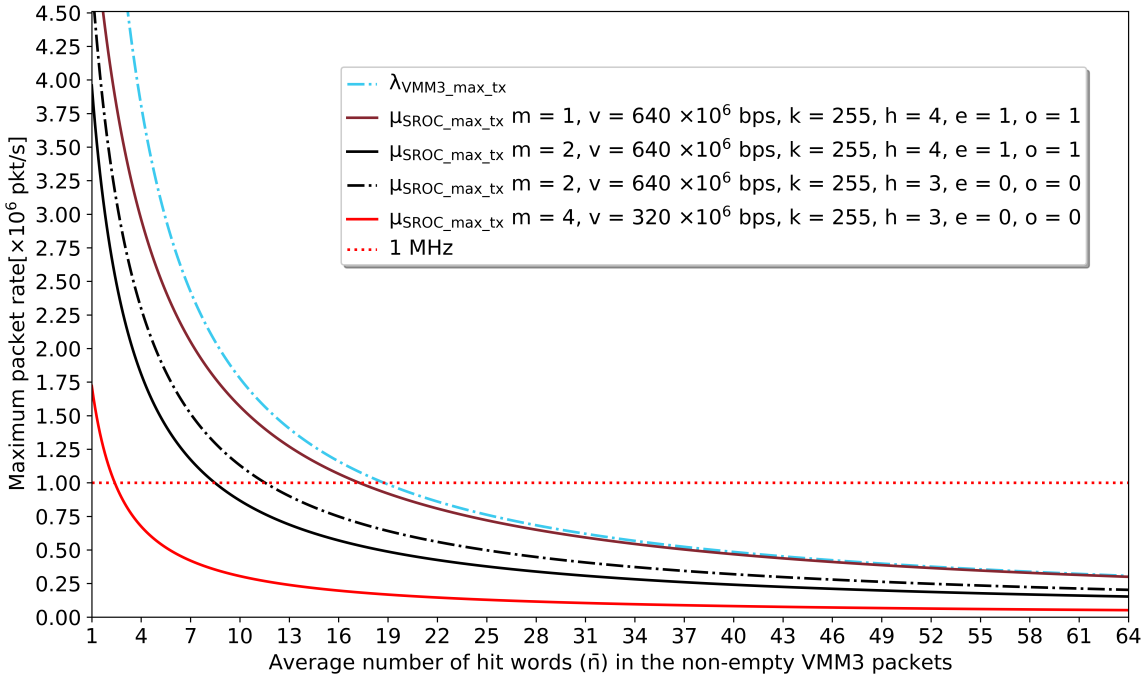


Figure 2.20: The maximum rates of packet transmission for the VMM3 and SROC as functions of $\bar{n}$, in different scenarios and considering $p = 20\%$.

2.4.4 SROC maximum processing rate

The maximum processing rate of the Packet Builder FSM is reached when the associated input FIFOs (i.e. TTC and VMM Capture) are not empty (i.e. they continuously have data to be dequeued) while the output SROC Packet FIFO is not full (i.e. it continuously has free space for data to be enqueued). The FSM is designed to enqueue the resulted 33-bit wide data words back-to-back (i.e. in consecutive BC clock cycles) when possible. Consequently, the 33-bit wide input data words are also dequeued back-to-back from the selected VMM Capture FIFO at the same time. However, in some states it is impossible to enqueue L1 data after a single clock cycle since it is either unavailable yet (e.g. while dequeuing an unselected L0 packet or changing the selected VMM Capture channel) or it is being constructed (in the case of the trailer word which includes the checksum field).

Considering the same descriptors for the input traffic from the previous section (i.e. p and n) and the formulas from Section 2.4.2, the average L0 packet occupies the following amount of addresses in the VMM Capture FIFO:

$$\text{size}_{addr\_avg\_L0\_pkt} = (1-p) \cdot (\bar{n}+1) + p$$

Considering m VMM Capture channels associated with the SROC, the resulted average L1 packet occupies the following amount of addresses in SROC Packet FIFO:

$$\text{size}_{addr\_avg\_L1\_pkt} = p^m + \sum_{i=1}^{m} \binom{m}{i} \cdot (1-p)^i \cdot p^{m-i} \cdot (2+\bar{n})$$

A normal system operation is assumed: no overflow flags or patterns from VMM3, $0 < sel \leqslant 1$, L1 triggers that match the L0 packets and no reason for sending *dummy* hits or set *VMM missing* flags.

The design of the SROC Packet Builder FSM has the following characteristics related to the processing speed:

1. When the empty signal of the TTC FIFO becomes low, it spends two BC clock cycles to dequeue the L1 trigger data and select the first VMM Capture.
2. From each selected VMM Capture channel it dequeues the L0 packets, one by one, until the L0 header matches or is older than the searched L1 trigger.
3. When dequeuing an L0 packet, one BC clock cycle is spent for checking the header data and then the occupied addresses are freed one by one in consecutive BC clock cycles. It does not matter if corresponding output data is produced and enqueued. Thus, the time for dequeuing an averaged sized L0 packet, in BC clock cycles is $\mathrm{t}_{dequeue\_avg\_L0\_pkt} = 1 + (1-p)\cdot(\bar{n}+1) + p$.
4. Between dequeuing L0 packets from the same selected VMM Capture channel, no additional BC clock cycles are spent.
5. When the selected VMM Capture changes to the next associated one, one BC clock cycle is spent.
6. After the last VMM Capture is processed, one BC clock cycle is spent.
7. Two BC clock cycles are spent preparing the trailer word. In the considered normal system operation, a trailer word is produced only when at least one L0 hit word is found. The probability that at least one matching L0 packet contains hit data is $\mathrm{P}_{any\_hit\_data} = 1 - \mathrm{P}_{no\_hit\_data} = 1 - p^m$
8. In the case when the L1 trigger matches only L0 null-events (i.e. with the probability p^m), an L1 null-event is formed. In case the ROC's configuration enables the transmission of L1 null-events (i.e. o), an additional BC clock cycle is spent enqueueing it.

Considering all these, the average time spent by the FSM processing an L1 trigger, in BC clock cycles, is:

$$\mathrm{t}_{proc\_L1\_trigg} = 2 + m\cdot\{\frac{1}{sel}\cdot[1+(1-p)\cdot(\bar{n}+1)+p] + 2\cdot(1-p^m) + o\cdot p^m + 1\} =$$

$$= 2 + m\cdot\{\frac{1}{sel}\cdot[(1-p)\cdot(\bar{n}+1)+p] + 2\cdot(1-p^m) + o\cdot p^m + 2\}$$

The maximum average processing rate of L1 triggers, in trigg/s, is:

$$\mu_{SROC\_max\_proc} = \frac{40\cdot 10^6}{2 + m\cdot\{\frac{1}{sel}\cdot[(1-p)\cdot(\bar{n}+1)+p] + 2\cdot(1-p^m) + o\cdot p^m + 2\}}$$

In Figure 2.21 the theoretical VMM3 and SROC maximum transmission packet rates are plotted as functions of $\bar{n}$ alongside the SROC's maximum average processing rate. In this case $p = 20\%$ and traces for all the SROC's possible transmission bit rates are depicted. The continuous blue trace represents the maximum average processing rate considering $sel = 100\%$. The maximum average processing rate does not keep up with the maximum average SROC transmission rate for $m = 2$ and small $\bar{n}$. The more frequent changing of the selected VMM Capture channel is the limiting factor. One should note that the processing logic uses the BC clock signal which is four times slower than the RO clock signal used for transmission and reception. The interrupted blue trace represents the maximum average processing rate considering $sel = 1\%$. Basically for each L1 trigger, an average of 100 L0 packets are dequeued from each of the two associated VMM Capture channels. Only 1% of the input data is transmitted to the output, thus, a transmission bit rate $v = 80\cdot 10^6$ bps is enough.

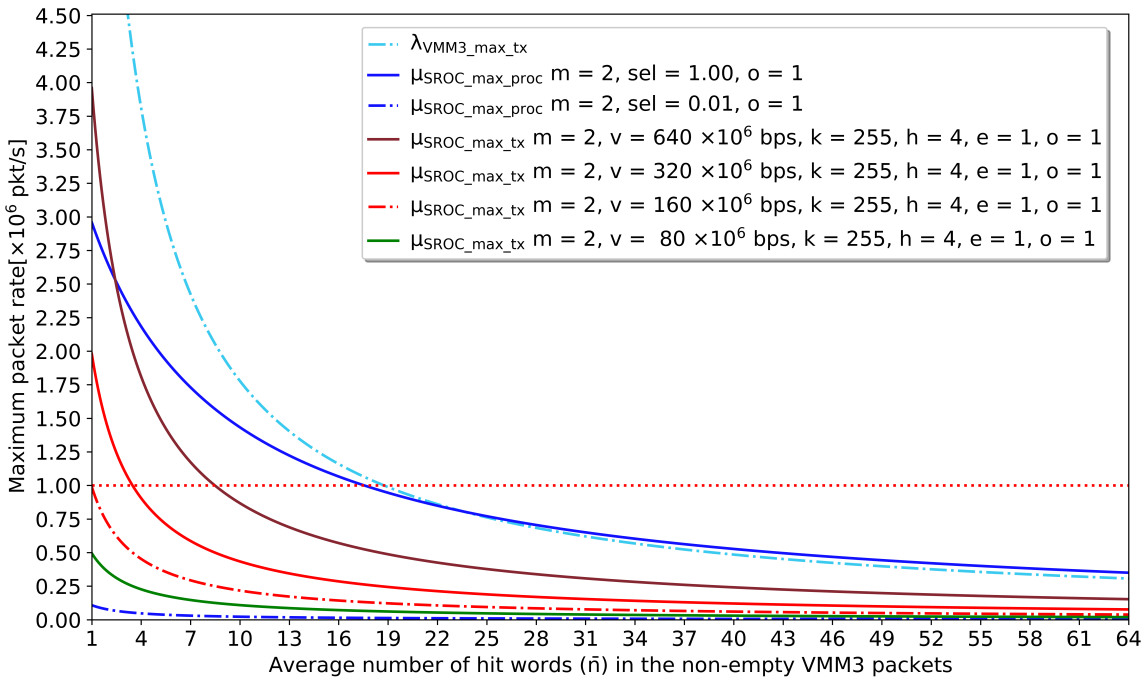


Figure 2.21: The maximum rates of packet processing and transmission as functions of $\bar{n}$, for all the possible SROC transmission bit rates and considering $p = 20\%$.

2.5 The SROC packet building algorithm

The SROC's packet builder FSM algorithm is detailed in Algorithms 4, 5, 7 and 6 from Appendix A.1. Within Algorithm 4, at lines 32 and 33 Algorithms 5 and 6 are called, respectively. In turn, Algorithm 5 calls Algorithm 7 at line 23. Thus, Algorithm 4 represents the main structure while Algorithms 5 and 6 represent the SROC behavior when it is in *normal* mode. Algorithm 5 depicts the processing of all the associated VMM Capture channels for one L1 trigger while Algorithm 6 presents the decisions and actions executed once all these channels are processed. Algorithm 7 is a detail within Algorithm 5 depicting what happens when the L0 OrbitID matches the L1 triggered OrbitID.

The instances of structures, arrays, variables and constants are written with *italic* style while the definitions of the functions use the ordinary style. Auxiliary tables that provide explanations are included. The called functions are described in Table A.1 while the comments are enumerated and expanded in Table A.2. The structures and types definitions are presented in Table A.3. The variables and constants that are not explained within the comments are described in Table A.4.

The transition graph of the resulting Mealy FSM is depicted in Figure A.1 from Appendix A.2. The schematic of the associated logic is shown in Figure A.1 from the same Appendix.

2.6 Conclusions of the ROC design

The ROC design was successfully signed-off and submitted on schedule to Metal Oxide Semiconductor Implementation Service (MOSIS) [160] to be integrated within an MPW, alongside the VMM3 and other NSW ASICs.

The design of the ROC's digital part implied the following procedures:

- establish the top-level architecture based upon the chip's context, its requirements and interfaces;
- establish the individual requirements and the detailed architectures of the modules and blocks from the top-level architecture;
- design the internal organization of each module and block, resulting in schematics of the digital logic, data formats, FSM transition graphs and tables (as presented for the Packet Builder FSM);
- describe each module and block in synthesizable RTL Verilog code;
- establish the functional features of each module and design dedicated exhaustive Verification Environments (VEs) for them, described also in Verilog but not necessarily synthesizable;
- verify each module in the corresponding dedicated VE through simulations, adjust its architecture and internal organization and correct the bugs from the RTL code and the VE accordingly, until the module is fully validated;
- integrate the validated modules and blocks into sub-groups and groups (also described in synthesizable RTL Verilog code) following a bottom-up strategy until the top module representing the entire ROC's digital part is complete;
- design exhaustive VEs, described in non-synthesizable Verilog code, for the resulted sub-groups and groups of modules;
- verify the sub-groups and groups of modules in the corresponding dedicated VEs through simulations, adjust their architecture and internal organization and correct the bugs in the RTL code and the VEs accordingly, until the RTL is fully validated;
- establish the synthesis design constraints (e.g. IO timings, characteristics of each clock signal, relations between clock domains, fan-outs, etc.) and express them as Synopsys Design Constraints (SDC) statements;
- import the validated RTL code, the design constraints and the libraries of the implementation technology into the synthesis tool and adjust the RTL (which then must be again verified) and/or the design constraints if errors and warnings are reported (e.g. incorrect inference of latches [46]);
- perform the synthesis of the design into a netlist of cells from the implementation technology libraries;
- solve every error and warning raised during the synthesis process (e.g. failed routes during Static Timing Analysis - STA) by adjusting the flexible constraints (e.g. fan-outs, IO timings for the phase-adjustable interfaces, etc.) and the RTL code (which then must be again verified);
- perform Logic Equivalence Check (LEC) between the resulted netlist and the RTL code once the synthesis process succeeds (i.e. all error, warning and information messages were fully solved or accounted for);
- solve any logical discrepancy reported by the LEC tool by adjusting the RTL code (which must be then re-verified and re-synthesized);
- run gate-level simulations of the synthesis netlist with back-annotated propagation delays (in the Standard Delay Format - SDF) in all the corner cases provided in the implementation technology libraries, using the adapted top-level VE;
- check if there are any logic differences between the RTL and the gate-level simulations (at this point there shouldn't be any as long as the verification,

synthesis and LEC were successful);
- check if any setup, hold and glitches violations are raised in the gate-level simulations; if found, change the phases of the input signals, especially the supplied clock signals, until none are raised (assuming the STA passed);
- establish the physical implementation constraints (SDC format) which are mainly derived from the synthesis constraints;
- import the synthesis netlist, the macroblocks (i.e. SRAMs), the implementation technology cell and physical libraries and the physical constraints into the physical design tool (i.e. floor-planning, Place and Route - PNR and sign-off/tape-out);
- perform the IO placement and the floor-planning;
- manually place the macroblocks on the core area and instruct the tool to place the rest of the cells from the imported synthesis netlist;
- inspect the reports generated by the tool: worst path delays, congestion, area, estimated power consumption, etc;
- solve every error and warning raised during the placement process (e.g. failed routes during STA) by adjusting the placement and the physical constraints (e.g. replication, fan-out, etc.);
- export the after-placement netlist and the updated SDF propagation delays;
- run gate-level simulations with the after-placement netlist and back-annotated updated propagation delays (SDF) in all the corner cases provided in the implementation technology libraries, within the adapted top-level VE;
- check if there are any logic differences between the RTL and the after-placement gate-level simulations (at this point there shouldn't be any as long as the placement was successful);
- check if any setup, hold or glitches violations are raised in the after-placement gate-level simulations; if found, vary the phases of the input signals, especially the supplied clock signals, until none are raised (assuming the STA passed);
- instruct the tool to perform the routing of the power and ground supply nets to every cell and macroblock;
- instruct the tool to perform the clock trees synthesis;
- instruct the tool to route the reset signals;
- instruct the tool to route the rest of the signals;
- inspect the reports generated by the tool: worst paths, congestion, area, estimated power consumption, power delivery, etc.;
- solve every error and warning raised during the routing process (e.g. failed routes during STA) by adjusting the placement, routing and the physical constraints (e.g. replication, fan-out, etc.);
- export the after-routing netlist and the updated SDF propagation delays;
- run gate-level simulations with the after-routing netlist with back-annotated updated propagation delays (SDF) in all the corner cases provided in the implementation technology libraries, within the adapted top-level VE;
- check if there are any logic differences between the RTL and the after-routing gate-level simulations (at this point there shouldn't be any as long as the routing was successful);
- check if any setup, hold or glitches violations are raised in the after-routing gate-level simulations; if found, vary the phases of the input signals, especially the supplied clock signals, until none are raised (assuming the STA passed);

- instruct the tool to perform Design Rule Check (DRC) and solve the reported violations (e.g. extend the below minimum area metallization surfaces);
- export the final netlist and propagation delays SDF file, run the final gate-level simulations and validate the design;
- export the final design as a GDSII file.

The digital logic from the analog part of the ROC, consisting of a register bank interfaced with an I^2C slave, is similar to the digital part's Config block. Thus, a similar design flow was performed. Three out of four ePLL macros were modified by an external (not from Transilvania University of Brașov) engineer. The ePLL macros were handled during the analog part development just as the SRAMs were during the design of the digital part: black boxes with behavioral Verilog models. The resulting analog and digital separate designs were interfaced together in a wrapper Verilog module that also contained the pad instances (also macro-blocks). This top design was verified, synthesized and implemented (considering the two sub-designs as macros). The final top-level GDSII design was submitted for fabrication.

Even if the designed and used VEs were described in non-synthesizable Verilog code instead of a higher-level Hardware Verification Language (HVL) like SystemVerilog [33], they employed pseudo-random data generation, writing results to and reading test data from files and included corner case tests. The resulted design validation led to a ROC design that meets all the established requirements of functionality, performance and area (see Chapters 3 and 4). The chosen FIFO sizes allow the ROC to cope with worst-case scenarios of data and trigger bursts and radiation-induced faults (see Chapters 3 and 4). The detailed queueing theory and mathematical models accurately describe its performance while the algorithm for the construction of the L1 packets describes its core functionality. The hardware implementation of the algorithm is depicted as a digital schematic associated with the FSM's transition graph rather than a complex text-only description.

For the verification process, simulations in which the (in development at that point) VMM3 L0 building RTL was interfaced with the ROC RTL were also performed. Small corrections to the requirements document were suggested. One example is the replacement of the 8b10b special symbol *K.28.7* with the *K.28.6* as the EOP to avoid the requirement of a complicated synchronization circuit since *K.28.7* can form false shifted *comma* symbols. Another example was the value of the *PATTERN* BCID which did not match between the VMM3 implementation (0xFE8) and the initial requirements document (0xEE8) so the value was updated in the latter and then used within ROC's logic.

Chapter 3

ROC Testing

In this chapter the FPGA-based test environment used for validating both the ROC's mathematical model and its digital design is detailed. Since the ROC ASIC also supplies clock and TTC signals with adjustable phases to the other context ASICs, a separate, also FPGA-based, test environment evaluates the ROC *analog* part. Both test environments were used for the quality control mass-testing of the fabricated ROC chips. A modified version of the digital test environment was used for the ROC radiation qualification tests which is the subject of Chapter 4. The emphasis of the current chapter is on the digital test environment because the author contributed exclusively to it.

The first section presents the main techniques for testing a digital ASIC. Testing differs from verification, even though in colloquial speech they are synonymous. The two notions are explained in the context of ASIC design. The second section of the chapter extensively presents the ROC digital test setup (top-view and detailed architecture) and includes an optimization method for its most critical part: correct calibration of the interfaces between the Device Under Test (DUT) and the Testing Device (TD). The DUT cannot be assessed correctly if the communication with the TD is defective. While the ROC adheres to the paradigms of a synchronous ASIC design, it contains asynchronous elements determined by its context and requirements. Both its testing and operational environments are asynchronous. The experimental result sub-section includes the digital design test coverage, the ROC's performance model validation, the real-world performance assessment for the optimized clock and data signals calibration method and the analysis of the mass-testing performed at Transilvania University of Brașov. The last section of the chapter depicts the conclusions related to the ROC testing. The research presented in this chapter was disseminated in [173] and [172].

3.1 ASIC Verification and Testing

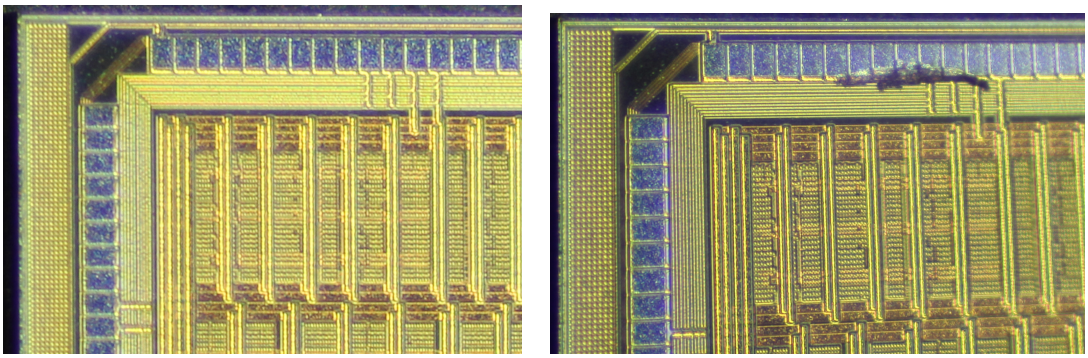
This section starts by explaining the distinct notions of verification and testing within the ASIC designing process. Next, the different ASIC testing types and techniques are presented. The rationale of why the FPGA technology is ideal for functional ASIC testing and verification is presented.

Verification is the process of validating the design of a chip before its manufacturing, to minimize the number of undiscovered bugs that could cause delays, redesigns and consequently high additional costs. Thus, it represents a mandatory quality assurance step within the developing process of an IC (Integrated Circuit). In the case of analog integrated designs, the verification consists of software simulations using mathematical models for the circuit elements and their connections (e.g. SPICE). For the digital designs, VEs are constructed and used in software simulations, with the aim of covering all the DUV's (Device Under Verification) functional features and their combinations. In these simulations, stimuli are injected in the DUV's inputs while its outputs are captured, analyzed and then compared with the expected response or with the outputs of a golden model block for the DUV. Both basic and advanced VEs can be described using Hardware Description Languages (HDLs, e.g. Verilog) with the added flexibility (compared to the DUV's RTL) that they do not have to be synthesizable. VEs can be described as software: a succession of instructions/operations, branch conditions and loops, operations with files and interactions with the user. The main disadvantage of using HDLs for constructing VEs is that they offer only basic operations and data types. It is similar to using the basic C language for the development of a complex program instead of C++ which supports for example Object-Oriented Programming (OOP) in conjunction with various libraries. Thus, Hardware Verification Languages (HVLs) that maintain the RTL modeling capability for a digital circuit but add flexibility when building VEs were designed and standardized by extending HDLs (e.g. SystemVerilog [33] which is based upon Verilog). They can employ paradigms like OOP for VEs, being closer to the high-level programming languages (e.g. C++, Python, Java, etc.).

The VEs can be also used to validate the netlists resulted from the synthesis, PNR and sign-off design implementation steps of the digital ASIC design, with or without back-annotated SDF files containing propagation delays for all paths and logic cells in different corner cases of process, supply voltage and temperature. A netlist is an ensemble of interconnected basic logic cells (from a library specific to the manufacturing technology) forming a circuit. It is expressed in textual form using a Hardware Description Language (HDL) and has very high complexity making it not easily human-readable/understandable. Thus, the HDL RTL circuits described by humans are considered behavioral (i.e. they describe the behavior of registers and the combinational logic between them) while netlists are structural models that (should) result in circuits with the same behavior. The simulations of the DUVs within their VEs are running far from real-time and imply approximation of the real world. As a reference, a simulation of 50 ms of operation of the sign-off netlist of the ROC digital part, with back-annotated delays, was achieved in approx. 24 hours on a computer with a 12-core Intel(R) Xeon(R) E5-2650 v4 Central Processing Unit (CPU) clocked at 2.2 GHz and 128 GB of DDR4 2400 MHz RAM.

Testing refers to the device validation post-manufacturing, i.e. its quality control. In contrast to verification, testing is applied to the actual hardware. It aims at discovering flaws produced in the manufacturing process, separating the manufactured samples into categories based upon their performance and/or usability (process called binning, the simplest of which is the separation of the *good* and *bad* devices) or validating that the device meets its specifications in the real world. Thus, there are two types of circuit testing: (i) post-fabrication and (ii) functional.

The post-fabrication testing assumes that the simulated (i.e. during verification) device is fault-free and uses it as a golden reference model in comparisons with the manufactured samples. It does not test its compliance with the specifications. In Figure 3.1 two microscope photos of the upper left corner (i.e. considering the same orientation as in Figure 2.17) of distinct real-world ROC dies, with (right) and without (left) physical defects, are illustrated. In the defective sample, several power and IO pads are affected. The post-fabrication testing must detect such defective chip samples. While a passed exhaustive/comprehensive verification procedure for a digital design, in conjunction with correct implementation, assures that an ideal manufactured sample of the device will operate correctly, bugs can still not be detected and thus corrected since simulations are approximations of reality and the development time is limited. Examples of problematic simulator issues are clock domain crossings [91], asynchronous inputs and meta-stability [118] [91] (i.e. unstable state between logic 1 and logic 0 caused by violated FF setup and/or hold time requirements). Furthermore, not all the possible scenarios that can happen in reality will be simulated during verification. This motivates the functional testing.



(a) Normal ROC die. (b) Defective ROC die.

Figure 3.1: Microscope photos (by Sorin Mărtoiu from IFIN-HH, Măgurele) of the upper left corner (as depicted in Figure 2.17) of two real-world ROC dies: one validated (a) and one defective (b).

In addition to a non-null probability of inducing defects, the manufacturing process has variability. This translates into the variability of the electrical characteristics of the produced samples (e.g. slew rates, fan-outs, delays, propagation times). This variability is considered during the implementation of the design (e.g. STA, gate-level simulations, SPICE) based upon the corner cases provided in the libraries of the implementation technology. The testing can further categorize the validated manufactured samples into classes of performance.

The Mean Time Between Failures (MTBF) represents the average time between failures of a system. What is considered a failure can vary, e.g. meta-stability for a flip-flop [118], SEU within a digital circuit (see Chapter 4), etc. The design and implementation of an IC should ideally aim at maximizing the MTBF. Its testing assesses and applies thresholds to the MTBF of the manufactured samples. During the quality control, the MTBF should be minimized since the per device testing time is many orders of magnitude smaller than the expected or desired MTBF. For ICs this means for example lower than nominal supply voltages, extreme temperatures, data rates and latencies above the maximum ones from their specification, worse

quality of the input signals, noisy or disruptive environments (see Chapter 4), etc. The key is where the threshold is set, i.e. what is considered a failure.

A relatively old (i.e. 1999) but comprehensive application note from Atmel [20] describing ASIC design guidelines, states that a *bad* device sample contains at least one internal net stuck at logic 0 or 1, regardless of the driving cell. This kind of fault is called stuck-at. The capability of a digital circuit to be tested easily is called testability and derives from other two factors: (i) controllability representing the ability to drive any internal net to both logic 0 and 1 (obviously not at the same time) using just the input signals and (ii) observability which is the ability to determine if an internal net is stuck in a certain state regardless of its driving logic value using just the chip's output signals. Within a digital circuit that has more than two pipe stages (i.e. more than two register levels on the path between its inputs and outputs) and a design without any consideration for its testability, only the first stage is directly controllable, only the last stage is directly observable and the middle stages are neither directly observable nor directly controllable. The testing process aims at covering as much as possible from the design. It is influenced by the variety of the applied stimuli (called test vectors). As a reference, an n-bit bus has 2^n possible states and $2^n \times (2^n - 1)$ possible transitions. The Atmel application note presents several techniques to increase the testability - Design For Testability (DFT) without also massively increasing the number of inputs and/or outputs pins or requiring many test vectors: break long counter or shift register chains, open feedback loops, include Built-In Self-Test (BIST) circuits within macroblocks, employ scan chains.

The first technique refers to long shift registers or chains of counters that increment or decrement relatively rarely during normal operation (e.g. where the overflow or underflow signal of one counter increments or decrements the next counter). It proposes the breaking of such chains by adding external test IO pins and internal logic. These additions allow the optional external drive and observation of the nets normally pulsed rarely. The second technique refers to registers containing feedback loops that determine their next value based upon their current state and other design signals. Similar to the first method, it suggests the addition of external test signals and logic. The next state can thus be optionally forced. BISTs are customized circuits added to complex macroblocks (e.g. memories, multipliers, etc.) that once enabled inject specific sequences of signals that influence all the internal nets and cells and thus produce a specific output in a certain number of clock signals. Any fabrication fault is revealed in this output.

The typical DFT method for digital ASICs is the scan design [215]. All the registers and FFs within a synchronous digital design contain additional logic and interconnections that organize them into long shift registers (called scan chains) controlled from outside while maintaining the design functionality. By putting the logic into test mode and shifting serial test vectors into these shift registers, one can put the entire design into a specific known state. For several clock cycles the logic is put back into functional mode. Then the logic is returned to test mode and the result is shifted out of the design and compared to a golden model. Scan chains can also debug designs by stopping the functional operation of the logic at any moment by enabling the testing mode and shifting the state out for inspection.

Joint Test Action Group (JTAG) [23] is a standard that extends the scan chain concept to the PCB/board level while maintaining the on-chip functionality. Multiple JTAG compatible devices (i.e. chips and/or modules inside the chips) are

daisy-chained together, being interfaced to a single JTAG interface. A reduced pin count JTAG version employs a star connection topology.

The FPGA is an IC technology which allows the devices to be configured after their manufacturing (in the field) with user-defined logic functions for hardware-accelerated tasks (e.g. testing). The ASIC represents the complementary IC technology that, once manufactured, cannot be reconfigured. An ASIC is customized for a specific utilization and is manufactured in large volumes while an FPGA has a more general purpose being intended for prototyping and low production volumes. The per-unit cost of an FPGA is larger than the one of an ASIC. An FPGA consumes more power than an ASIC for the same functionality. It cannot reach the same operating frequencies (i.e. lower performance) but it is more flexible. The typical resources available on an FPGA device for implementation of a digital design are: Look-Up Tables (LUTs) which emulate combinational logic, full adders since the addition/subtraction operations are frequent, multipliers, multiplexers for selection, configurable signal routing, Input-Output Blocks (IOBs) for interfacing with the outside world, FFs and registers for storing the results and states, PLLs for generation of clock signals, blocks of SRAM (e.g. Block BRAM - BRAM in Xilinx FPGAs) used to buffer and cross data between clock domains and transceivers for high-speed interfaces. FPGA devices are usually programmed through JTAG.

Often the two ASIC testing types - post-fabrication and functional - are combined. The functional testing may be able to toggle and observe all or close to all the device's internal nets. Thus, in the end, the functional test setup represents a hardware-implemented version of the VE. The FPGA technology is ideal for this task. While it is more difficult to design and implement a functional test setup (i.e. VE) on an FPGA because it must be synthesizable and implementable on that device, it offers advantages like: being able to observe several orders of magnitude larger operational times of the DUT, simulation times close to real-time and physical proof of concept setup for the device.

Depending on the design's requirements and the available FPGA devices, the VE development could be done directly for the FPGA in conjunction with the development of the entire or partial DUV on the same FPGA. The two FPGA design parts could be easily interfaced using the FPGA configurable routing resources. Rather than applying and using assertion statements (i.e. logic statements that verify properties or specific conditions) described using an assertion specification language (e.g. SVA - SystemVerilog Assertion), these can be easily described as synthesizable modules (i.e. checker modules). Through the FPGA synthesis and implementation, the DUV's RTL can also be checked for design bugs that are usually not caught by the HDL simulator compiler (e.g. inference of latches [46] instead of purely combinational logic for an always block with at least one missing item in its sensibility list). Observability within the FPGA-based verification environment can be achieved using Integrated Logic Analyzers (ILAs) - see Chapter 5.

Wafer testing is a step executed before the mounting and die slicing procedures during the semiconductor device fabrication. An automatic test equipment called wafer prober tests all the ICs present on the wafer, by applying test vectors. The device is able to align its probing tips with sufficient precision on the ICs pads.

The functional tests are the most critical for the success of an IC because they assure their operation in real life [142]. Their thoroughness determines the quality of the validated samples but implies more time and a lower yield.

3.2 The Quality-Control Digital ROC Test

The ROC chip required exhaustive functional testing with coverage of all its features, FSMs states, every bit from all its SRAMs, the assessment of the data processing performance and the characterization of the supplied clock and TTC command signals and their phase-adjustment. Based upon the specifications (Chapter 2, Section 2.1) and the developed and used VE, a testing checklist was synthesized and then enlarged and approved by the CERN NSW electronics group. The most challenging requirements were: generation of L0 packets with randomized content, rate and size; average L1 trigger and L0 packet rates configurable in steps of 100 kHz up to 1 MHz; configurable L1 trigger latency in steps of 10 μs up to 60 μs; emulation of data and L0 trigger queuing in VMM3; bandwidth saturation for all the SROC output lanes; coverage of all flow and congestion control mechanisms; production of worst-case L0, L1 and VMM3 packet bursts in a loop; coverage of all the configuration for the ePLLs and the quality assessment of the forwarded TTC commands and clock signals. The work was split and two separate functional test setups were developed in parallel: one for testing the ROC digital part which is the focus of this chapter and the second one for the ROC analog part which is very briefly presented in what follows. The analog testing assesses the quality and phase-programmability of the internal and external clock signals (i.e. frequency, duty-cycle, period jitter, phase in relation to the reference), covers the settings for each ePLL and assesses the phase-programmability and validates the correct forwarding of the decoded TTC commands relevant to the other NSW ASICs. The accurate decoding of these commands from the TTC stream is covered by the digital testing.

A synchronous ASIC design is the safest design from the time-domain point of view [20] [46] [24]. Asynchronous designs are to be avoided as much as possible. According to [20], to be considered synchronous, a design must respect the following two conditions : (i) all of its sequential elements must be paced by a *true* clock signal and (ii) all the sequential elements are sensitive to the same edge of the same clock signal having the same phase everywhere in the design.

The first condition refers to the fact that the clock signal pacing each storage element must not be derived from the data output of any other sequential element or does not pass through any combinational logic. The ROC design respects this condition except into the actual ePLL macro-blocks where the output clocks are fed through delay lines and multiplexers to achieve phase-programmability. A modern technique for reducing the power draw of digital ICs is clock gating: interruption of the clock signal fed to a sequential element when the stored value does not change (the input equals the current state) to reduce the dynamic power consumption to 0 (only leakage currents remain) [161]. This technique contradicts the first condition for a synchronous circuit. Some implementation technology libraries contain FFs and/or latches with built-in clock gating.

The ROC does not meet the second condition for synchronicity for the following three reasons: (i) its internal logic is paced by multiple clock domains that may or may not be in phase, (ii) it contains sequential elements that work on the positive edge of the clock while others are sensitive to the negative edge of the clock (the incoming VMM3 data, the TTC stream and the output data are DDR), (iii) the internal clock signals do not have the same phase for every paced sequential element due to different skews and propagation delays caused by the clock trees. The last

reason is true for every digital IC in the real world [25]. The clock tree synthesis is a distinct step within the implementation of an ASIC, usually performed automatically, based upon the design constraints, as part of the PNR procedure. [20] and [25] specify that a solution for achieving the simultaneous administration of a single clock signal to all sequential elements within a design is a clock tree with balanced fan-out and equal depth to the leaf nodes (i.e. application points). In practice, the propagation times for a clock signal to the sequential elements are allowed to have some skew to relax the constraints applied to the PNR tool. Thus, the *set_clock_uncertainty* SDC sentence includes the expected setup and hold clock uncertainty resulted from the expected maximum jitter and skew. These constraints affect the STA but it is assumed that if the PNR implementation step is reached the circuit passed the synthesis STA.

DDR is often used to increase the rates of data throughput (e.g. DDR DRAM - Dynamic RAM). Most of the digital circuits use different clock signals simultaneously (e.g. the 8-bit Atmel AVR ATmega16 MCU (Microcontroller Unit) employs CPU, IO, Flash, Timer and ADC distinct clock domains [3]). Thus, if one follows the second condition for synchronicity to the letter it concludes that most digital circuits are asynchronous even if their operation is dictated by clock signals.

The first condition for a synchronous design from [20] represents its complete definition in [46]. All the activity within a synchronous design is controlled by clock signals. The data inputs of all the contained registers and flip-flops are sampled on the active edge of the pacing clock signal and become the new states which are transferred to the outputs. The outputs drive combinational logic which ideally should determine the next states for the following registers and flip-flops before the next active edge of the clock signal. Thus during the synthesis and implementation, STA is performed for synchronous designs (see Chapter 2, Section 2.6). A synchronous design does not contain combinational loops, asynchronous pulse generators or rely on delay chains/lines. From this point of view, the ROC is synchronous except it contains delay lines for the externally supplied clock signals and TTC commands.

The ROC context is considered asynchronous because its components (e.g. the VMM3s associated with one ROC) have distinct clock domains and delay lines are used (e.g. within ROC and GBTx).

The ROC is mostly synchronous. It contains asynchronous elements: its FIFOs, the TTC Capture module, delay lines for the forwarded clock signals and TTC commands. It does not contain built-in automatic phase-detection circuits for the incoming data so it requires a more complex functional digital test setup and a more complicated set-up procedure once it is installed on the NSW FEBs.

3.2.1 Digital test setup top-view

For validating the digital design and mass-test the ROC, a generic but flexible test setup architecture was embraced. The overview of this test setup is described in [175]. The test setup top-view for a general digital chip with IO channels is described and then the adjusted one for the ROC is depicted.

The considered general test setup contains configurable generators of stimuli that are connected to the DUT's inputs, output data monitoring modules interfaced to each DUT output and a unit for status checking and control, as depicted in Figure 3.2. The generators allow the creation of various scenarios that cover the

DUT's functional features. The output monitor modules validate the used protocol and the coherency of the entire output data in correlation with the used configuration for the generators. The test setup also assures the supply of power and the required clock and reset signals for the DUT. The status and control unit manages the testing procedure. In Figure 3.2 three types of communication lines between the testing device and the DUT are emphasized: (i) DUT input channels which are interfaced to pattern generators from the testing device, (ii) DUT output channels which are assessed using interface monitors in the testing device and (iii) bidirectional channels. In the case of a bidirectional DUT channel, both a generator and a monitor are used inside the testing device (e.g. an I^2C interface).

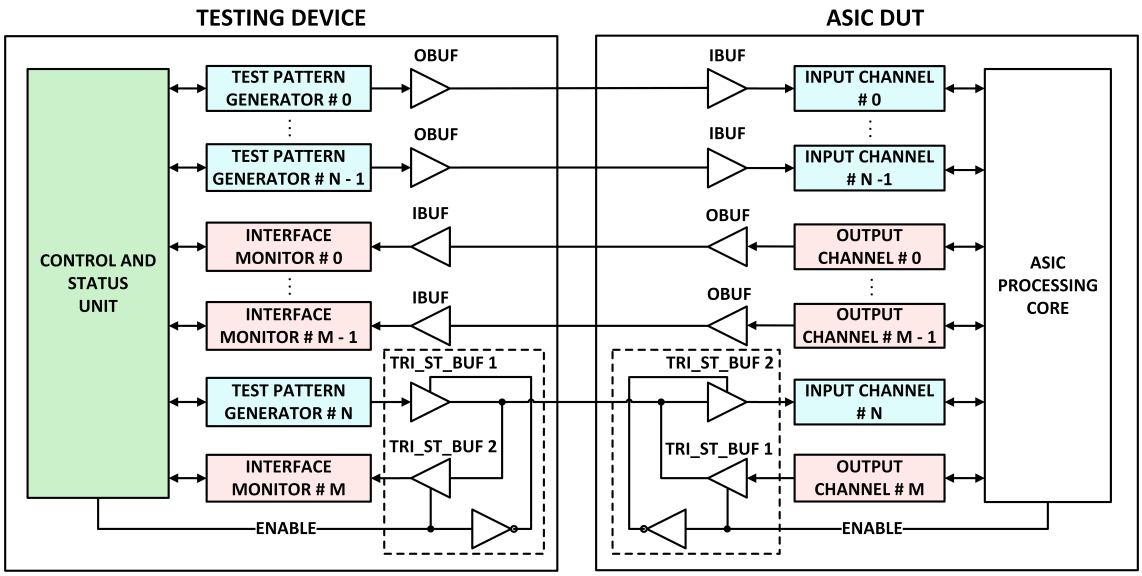


Figure 3.2: The embraced flexible general test setup architecture for a digital chip with IO channels.

The customized ROC test setup is based on the Xilinx Kintex Ultrascale KCU105 FPGA evaluation board [209] and custom accommodating PCBs for the ROC. The input data generators, the output monitors modules and the status and control unit are implemented on the FPGA, as firmware. The firmware top-view architecture is detailed in Figure 3.3. The DUT's 8b10b serial input channels are the eight VMM Capture channels. The ROC's TTC Capture module represents the DUT's control serial input channel, while the DUT's 8b10b serial output channels are the four SROCs. L0 packet generators reproduce the VMM3 data streams. Based upon the injected data and the test configuration, a separate generator outputs the TTC stream. The configuration and monitoring of the ROC's digital and analog parts is achieved through two I^2C master modules. The output data analyzers check the encoding and coherency of theROC output data streams and the parity bits, checksum fields and content of the L1 events. A Xilinx MicroBlaze [212] 32-bit Reduced Instruction Set Computer (RISC) soft-core microprocessor is instantiated in the FPGA design and runs a custom software that controls and monitors the other modules. Several peripherals are associated with the processor and are used for debugging, communication with the user, configuration of the custom PCB and precise timing. The clock signals that pace the FPGA design and the ones supplied to the ROC are generated by a PLL. The locking signal of this PLL is used for

the generation of reset signals. A Xilinx Integrated Logic Analyzer (ILA) [206] increases the observability within the FPGA design by sampling relevant signals and transferring the resulted data to a computer through JTAG for analysis. The custom PCB assures the interconnection between the ROC and the FPGA through Versa Module Europa or Versa Module Eurocard bus (VMEbus) International Trade Association (VITA) 57.1 FPGA Mezzanine Card (FMC) connectors [209] [50]. The next sub-sections elaborate on every element of the test environment.

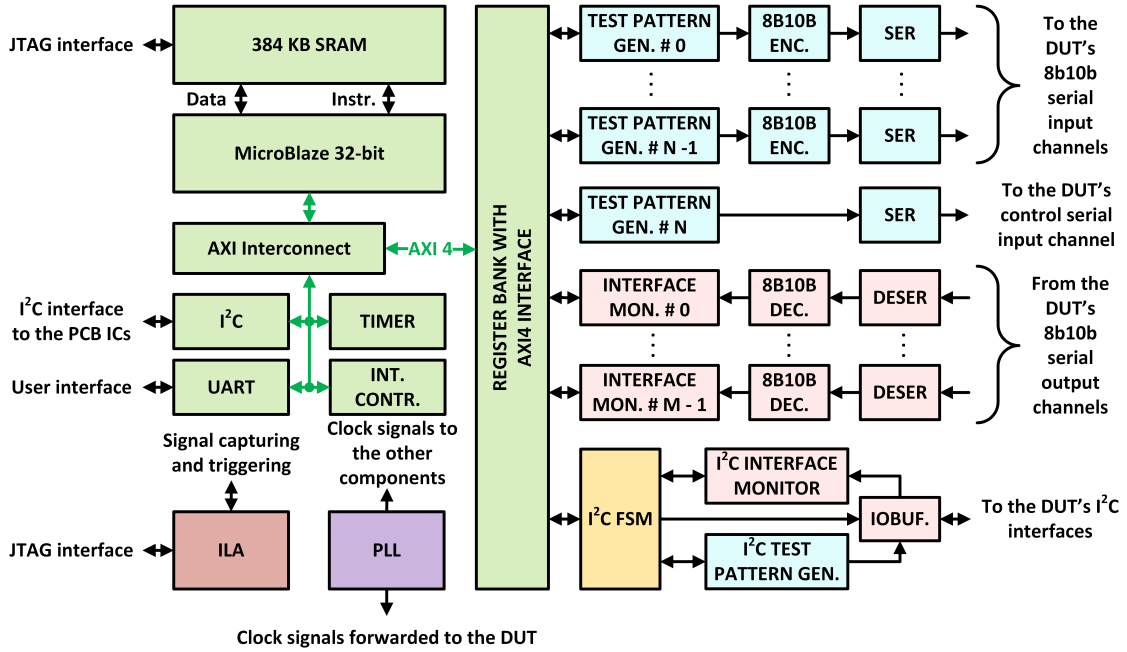


Figure 3.3: The top-view architecture of the ROC digital functional test setup.

3.2.2 The architecture of the proposed ROC functional digital test setup

The detailed architecture of the ROC's functional digital test setup is presented in Figure 3.4. Elements from the FPGA deisgn, the custom accommodating PCBs and the FPGA evaluation board are illustrated. The emphasis is on the clock domains, reset signals and data flows. The white square shown in the center of the *Test-Bench sub-system* is the ROC DUT having the same topology as in Figure 2.1. Each test setup component is detailed in the following sub-sections.

Testing PCBs

Five versions of ROC testing PCBs were designed, implemented and used. The first four differ mainly by how the ROC package evolved: (i) with the ROC wire bonded directly to the PCB (illustrated in Figure 3.5a), (ii) with the ROC as QFP 144 mounted on the PCB (illustrated in Figure 3.5b) and with the ROC as BGA 16×16 package either (iii) surface mounted (detailed in Figure 3.5c) or (iv) placed within an Ironwood socket for chip interchangeability (depicted in Figure 3.5d). In all four figures the single High Pin Count (HPC) FMC connector is situated at the bottom. The fifth PCB type (detailed in Figure 3.6) uses a different BGA socket for

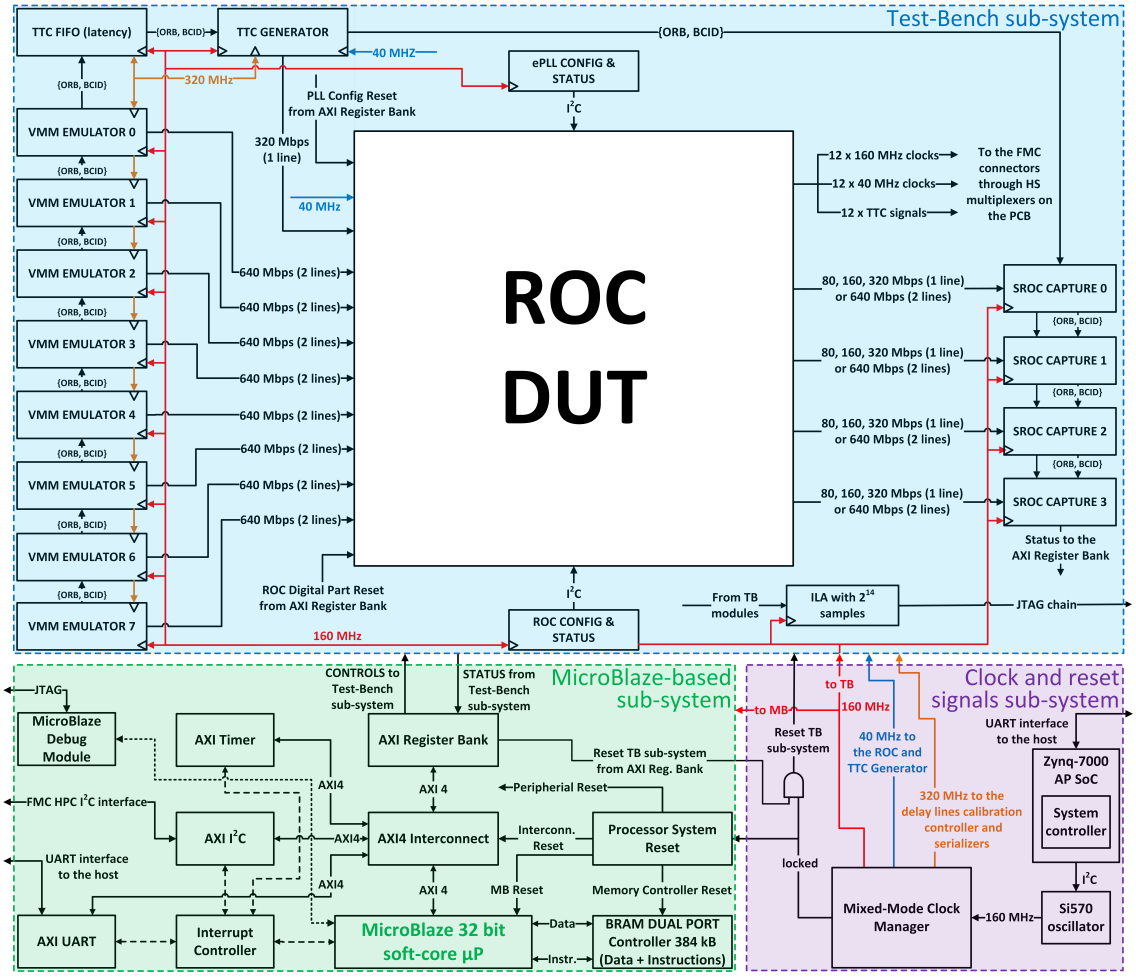
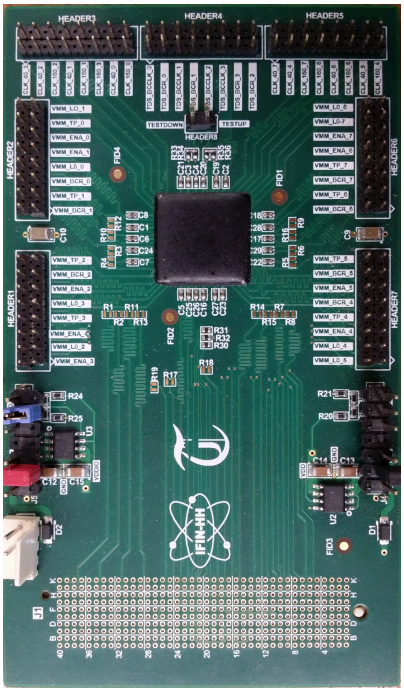


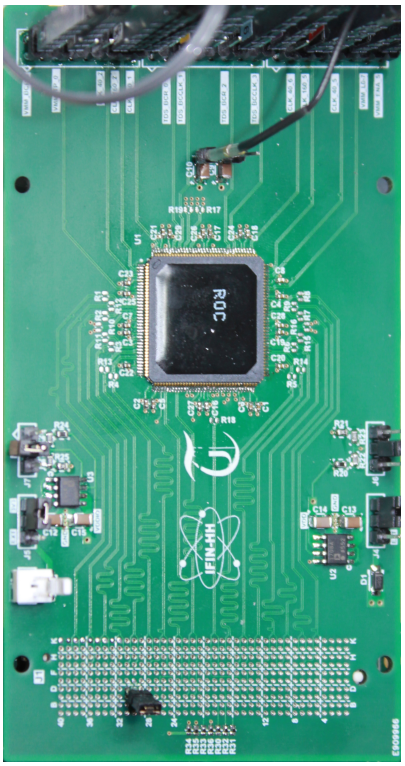
Figure 3.4: The architecture of the digital test setup, emphasizing the clock and reset signals and the data flows.

easy and rapid insertion/removal of the tested ASIC, different and supplementary ICs and the second FMC connector which is Low Pin Count (LPC). The signals mapped to the HPC FMC connector are kept on the same pins for all PCB versions to maintain firmware compatibility. In Figure 3.5a the area where the ROC die is placed and wire-bonded is covered with a low-viscosity polymer resin for protection. The same area was depicted uncovered in Figure 2.18a. The ROC in the QFP 144 package provides only a subset of the chip's IO pads: some of the output clock signals and TTC commands are not connected. All the signals related to the digital packet processing logic are supplied to the FMC connector.

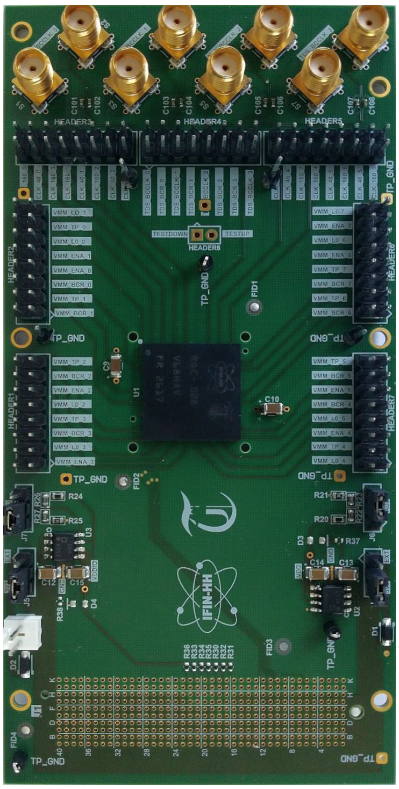
On all the board versions the ROC is powered by two LDO voltage regulators, one supplying the ROC IO pads and the other one the internal ROC core logic. In the first four PCB versions, the output voltages of the regulators are selectable between 1.1 V, 1.2 V (nominal) and 1.3 V using jumpers. The regulators' input voltage is selectable between an external source or the 3.3 V supplied by the FPGA evaluation board used for testing through the FMC connectors. All the serial data links coming into or out of the ROC, the reference and forwarded clock signals and the decoded TTC commands use the SLVS standard which requires a termination resistance of 100 Ω at the receiving end. For the SLVS signals which are received by the FPGA the line termination is made inside the FPGA pad. For the SLVS signals



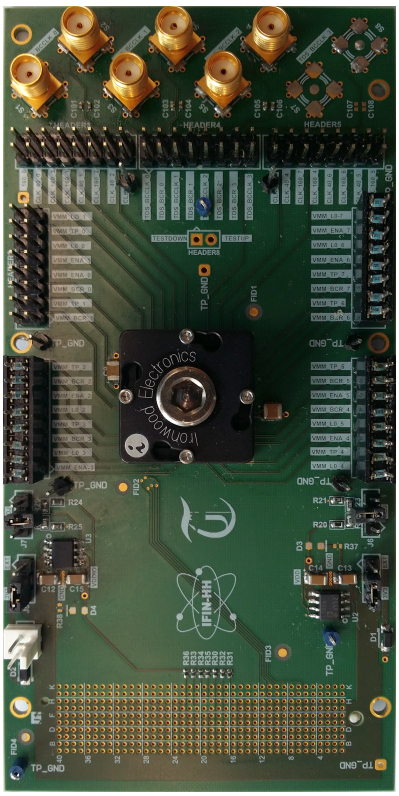
(a) The first version with the ROC die wire bonded and protected with a low-viscosity polymer resin (the ROC die is depicted in Figure 2.18a).



(b) The second version with the QFP 144 ROC surface mounted.



(c) The third version with the BGA ROC surface mounted.



(d) The fourth version with the BGA ROC placed within a closed-top socket.

Figure 3.5: The first four versions of the ROC testing PCB.

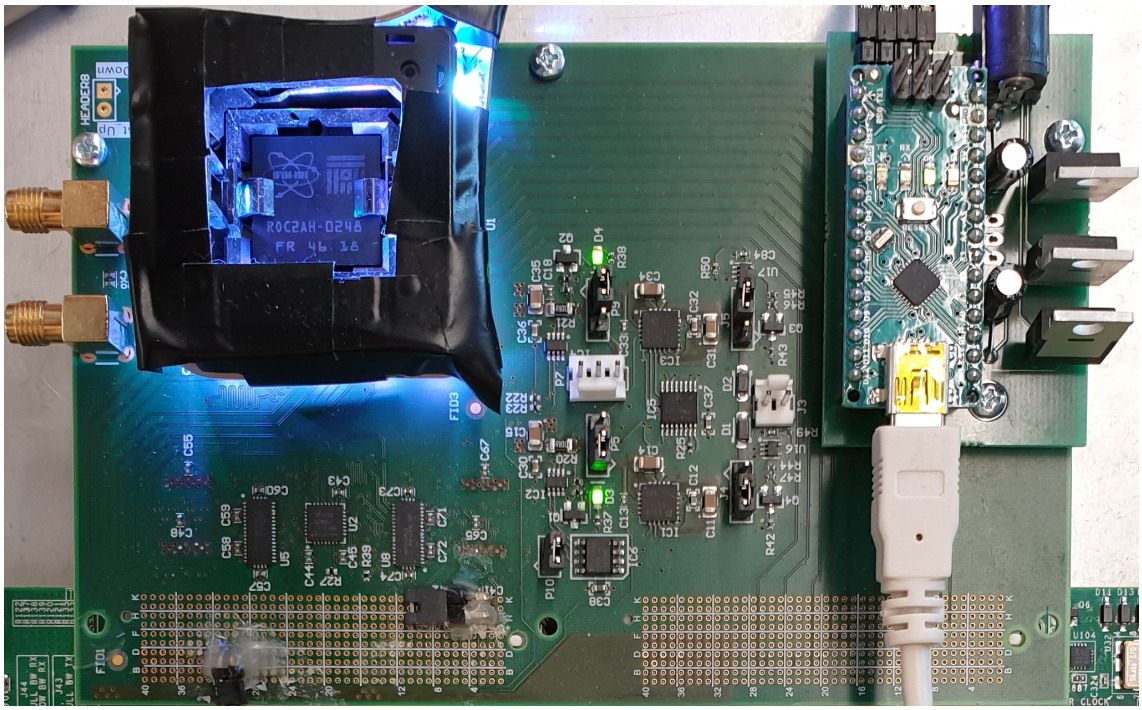


Figure 3.6: The final ROC testing PCB version with a BGA ROC placed in the open-top socket.

that are routed to external measuring points there is no termination on the boards. The ROC input SLVS lines are terminated inside the ROC input pads. Therefore no 100 Ω termination resistors are used on the testing boards. The single-ended signals are either pulled-up or pulled-down depending on the polarity e.g. the Serial Clock Line (SCL) and Serial Data Line (SDA) lines of the two ROC I^2C interfaces are pulled-up using 4.7 kΩ resistors.

The phase-adjustable clock signals generated by the ROC ePLLs and forwarded to the other ICs in the NSW front-end system and the associated decoded TTC commands are routed to pin headers in the first four versions of the testing PCB to be measured with an oscilloscope or logic analyzer. In the third and fourth versions of the testing PCB some ROC output clock signals are also routed to SubMiniature version A (SMA) connectors as shown in Figures 3.5c and 3.5d, topside, to be evaluated more precisely.

The chosen FPGA evaluation board provides two FMC connectors: one HPC with all 400 pins populated which is used by all versions of the testing board as the interface to the FPGA and one LPC with only 160 pins populated that is used only by the fifth PCB version as a secondary interface. Some of the FMC pins are interfaced directly with the FPGA chip, others are used for power supply, some are interfaced with other ICs from the evaluation board and some are not connected.

The fifth board version uses an open-top socket for fast insertion to mass-test the BGA packaged ROC, uses programmable regulators, current and power monitors and the second FMC connector to which the ROC output clock and TTC signals are routed through programmable high-speed SLVS switches. All these programmable ICs are connected to the dedicated I^2C bus of the HPC FMC connector which is linked with the I^2C master associated with the MicroBlaze microcontroller in Figures 3.3 and 3.4. In Figure 3.6 the fifth version of the ROC testing board is depicted

being powered on while interfaced with the FPGA evaluation board. The Arduino Nano board from the top right corner is used only for controlling the RGB LED strip that illuminates the socket area for easier chip sample identification. The Arduino board has no connection with the ROC testing PCB.

Digital testing FPGA design

The digital testing FPGA design has three main parts: (1) the *clock and reset signals sub-system*; (2) the *MicroBlaze-based sub-system* and (3) the *test bench sub-system*. The *clock and reset signals sub-system* supplies the necessary clock, reset and PLL locked signals. The *MicroBlaze-based sub-system* provides Universal Asynchronous Receiver-Transmitter (UART), JTAG and I^2C interfaces and manages the rest of the FPGA design and the ICs from the testing PCB. The *test bench sub-system* contains eight VMM3 data generators for the VMM Capture channels, one TTC generator for sending synchronization and trigger commands to the ROC, four output data analyzers for the SROCs, two I^2C masters for independently accessing the register banks of the digital and analog parts and FIFOs emulating the latency of the TTC commands and buffering the expected output BCs.

The clock and reset signals sub-system The FPGA evaluation board contains a Si570 oscillator [138], configured to supply a 160 MHz clock signal to the FPGA device. Its data-sheet states a peak-to-peak period jitter of 14 ps. Using this clock signal as a reference, a Mixed-Mode Clock Manager (MMCM) [202] within the FPGA outputs three in-phase clock signals of 320, 160 and 40 MHz, with the peak-to-peak jitter estimation of 94.983 ps, 108.43 ps and 143.129 ps, respectively. Both the *MicroBlaze-based* and *test bench* sub-systems use the 160 MHz output clock signals. The clock signal is also supplied to the ROC which uses it in case its ePLLs are bypassed. The serializers for the emulated L0 data and TTC stream use the 320 MHz clock signal. The delay lines used for the serial communication with the ROC are continuously calibrated for a steady propagation delay over temperature and voltage variations by IDELAYCTRL [203] Xilinx primitives. These primitives are also paced by the 320 MHz clock signal. The 40 MHz clock signal is supplied to the ROC as its reference BC clock signal. The order of the bits within the generated TTC stream is established based on this clock signal. The MMCM signals when its output clock signals are stable and aligned with an active high locked signal. This signal is used to reset the *MicroBlaze-based* and *test bench* sub-systems. All the reset signals in the FPGA design are active low, enabled asynchronously and disabled synchronously using the method described in [93] to avoid race condition with the clock signals. The sub-system is depicted in Figure 3.4 on a lilac background where it includes the oscillator and the controlling System On a Chip (SOC). The GBTx ASIC supplies the 40 MHz reference BC clock signal to the ROC in the NSW TDAQ system (see Chapter 2). The GBTx output BC clock signal has a measured 70 ps peak-to-peak jitter [201] which is half the jitter of the clock signal forwarded by the digital test setup. Nevertheless, the FPGA generated clock signal is suitable for testing since within the NSW the operation conditions, the GBTx clock signal will have increased jitter.

The MicroBlaze-based sub-system MicroBlaze [212] is a configurable soft-core RISC microprocessor designed by Xilinx. In the digital test setup, a 32-bit MicroBlaze was instantiated, configured and interfaced with 384 KB of BRAM for data and instructions, a JTAG debug module [213], an interrupt controller and an AXI Interconnect core [207] for managing memory-mapped slave devices. The following peripherals were also instantiated, configured and interfaced: AXI UART Lite [208], AXI I^2C [204], AXI Timer [205] and a custom AXI Register Bank. The AXI UART core represents the main communication channel to the host computer. The AXI I^2C core is employed to get access to the I^2C interface of the HPC FMC connector and thus reach the ICs on the fifth version of the testing PCB. The AXI Timer measures execution times and triggers operations. The custom AXI Register Bank consist of 2^7 32-bit wide memory-mapped registers used for interfacing the microcontroller to the *test-bench sub-system*. Some of the registers are driving input signals to the *test-bench* and can be read or written by the microprocessor. The inputs of the other registers are driven by *test-bench* output signals representing read-only status and error flags. A reset module establishes the proper timing for the enabling and disabling of the reset signals for all the peripherals and the microprocessor. All these modules form the *MicroBlaze-based sub-system* which is illustrated on a green background in Figure 3.4.

The test bench sub-system The test bench sub-system contains: (i) eight *VMM3 emulators*, one for each VMM Capture channel; (ii) one *TTC FIFO* that buffers the BC information of the generated L0 packets to emulate the latency of the corresponding L1 triggers; (iii) one *TTC generator* that outputs the appropriate TTC stream; (iv) four ROC L1 data analyzers called *SROC Capture* and (v) two *I^2C masters* used to independently access the configuration and status registers within both the digital and analog parts. In Figure 3.4 this sub-system is detailed on a blue background. Everything is custom-designed, except for the FIFOs and the Xilinx primitives used for serialization, deserialization and phase calibration of the IO signals.

The architecture of the *VMM3 emulator* is depicted in Figure 3.7. The *Packet Descriptor* module provides the following packet descriptors to the *Generator FSM*: the BC information to be put in the header as an increment relative to the previous packet, the number of *comma* symbols after the packet and the packet size in number of hit words. Except for the BC increment, the descriptors are different for each *VMM3 emulator*.

The average rate of packets is adjustable from 100 kHz to 1.4 MHz in steps of 100 kHz while the considered maximum L0 rate within the NSW context is 1 MHz. For each rate, predefined scenarios with various average percentages of empty packets and average sizes for the non-empty packets can be selected. In total there are 2^7 scenarios, each containing lists of 2^8 pseudo-random values for each descriptor. These lists are looped as the *VMM3 emulator* outputs packets. The queuing of the BC selection commands inside the VMM3 is emulated. Constant adjustable rate and size for the L0 packets can also be used.

The ATLAS Front-End Interface Requirements document [177] specifies the maximum theoretical L0A & L1A bursts for the trigger commands (consequently the bursts of L0 and L1 packets assuming no overflows). In a single-stage hardware

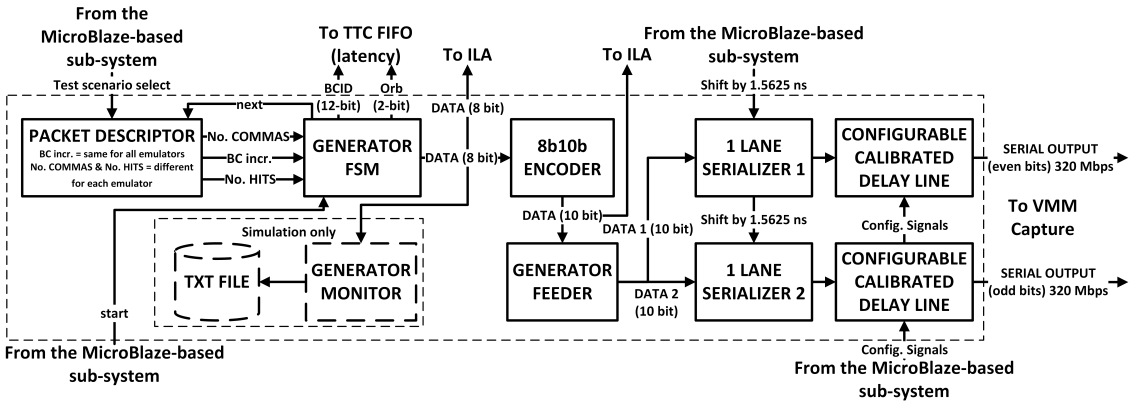


Figure 3.7: The VMM3 emulator architecture.

trigger, the following three statements apply to both L0A & L1A:

1. maximum four BCs can be selected from any five consecutive ones;
2. maximum eight BCs can be selected from any 20 consecutive ones (0.5 μs);
3. maximum 128 BCs can be selected in any 90 μs interval.

In a double stage hardware trigger, for the L0A only the third statement changes to maximum 128 BCs can be selected in any 30 μs time interval. For the L1A all three statements remain unchanged.

These burst parameters are declared as not yet finalized. For reference, during the ROC's design, implementation and testing it was considered that there will be maximum 128 selected BCs in any 128 μs interval in conjunction with the first two statements. Thus, the considered worst-case trigger (and packet) burst in a loop was achieved by following Algorithm 1. This algorithm was implemented as a distinct scenario in the *Packet Descriptor* module and was used in the performance assessment and validation of the mathematical model of the chip proposed in Chapter 2, Section 2.4. The model also shows that the ROC supports packet and selection rates above 1 MHz without data loss. This is proved by the mass-testing configurations from Table 3.3 and the presented results from Section 3.2.4.

Algorithm 1 Pseudo-code for the considered worst-case trigger and packet burst in a loop scenario.

```
1:  while True do                                  ▷ forever; 128 selected BCs in 128 μs
2:      for 16 times do                            ▷ i.e. 128 selected BCs in 8 μs
3:          for 4 times do                         ▷ i.e. 4 consecutive selected BCs
4:              select(current_BC)
5:              current_BC ← current_BC + 1
6:          current_BC ← current_BC + 1            ▷ no selection
7:          for 4 times do                         ▷ i.e. 4 consecutive selected BCs
8:              select(current_BC)
9:              current_BC ← current_BC + 1
10:         for 11 times do                        ▷ no selection
11:             current_BC ← current_BC + 1
12:     for 4800 times do                          ▷ i.e. 120 μs - no selection
13:         current_BC ← current_BC + 1
```

All the parameters for the triggers are summarized in Table 3.1.

Document	**VMM3-ROC Req. [123]**			**ATLAS TDAQ Req. [177]**		
Scenario	**Single**	**Double**		**Single**	**Double**	
Trigger	**L0/L1**	**L0**	**L1**	**L0/L1**	**L0**	**L1**
Max. avg. rate [MHz]	1	1	$\leqslant 1$	1	4	0.8
Max. latency [μs]	10	10	60	10	10	35
Cons. triggers [# L0/L1 in # cons. BCs]				$\leqslant 4$ in 5	$\leqslant 4$ in 5	$\leqslant 4$ in 5
Burst [# trigg. / # μs]				$\leqslant 8/0.5$ $\leqslant 128/90$	$\leqslant 8/0.5$ $\leqslant 128/30$	$\leqslant 8/0.5$ $\leqslant 128/90$

Table 3.1: Triggers parameters in both single and double scenarios extracted from the VMM3-ROC Requirements and the ATLAS Interface Requirements documents.

The Generator FSM creates the uncoded VMM3 data stream based upon the descriptors supplied by the *Packet Descriptor* and outputs it one byte at a time. The fields of the hit words are filled with individual counter-generated values, incremented for each new hit word. For each *VMM3 emulator*, the starting and the increment values of these counters are different. There is the option of using constant value hits (i.e. option used for the radiation test setup as described in Chapter 4). The generated bytes are 8b10b encoded, the odd and even bits of the resulting 10-bit symbols being fed to separate serializers paced by the 320 MHz MMCM clock signal. There is the option of delaying the streams with half a clock cycle (i.e. 1.5625 ns) to extend the coverage of the configurable delay lines. The BC information is pushed into the *TTC latency FIFO* and then used by the *TTC Generator* to send selection commands that match the transmitted BC information from the VMM3 packets. A *Generator Monitor* logs the generated uncoded data into a text file stored on the host computer, for debugging purposes, while simulating the RTL code of the test setup in conjunction with the ROC's digital part RTL or netlist.

The TTC Generator produces the TTC stream. It synchronizes the BCID and OrbitID counters from the ROC's TTC Capture module with its counters by issuing an OCR command. When the BC counters values match the BC information within the header of the first sent L0 packet (available from the TTC latency FIFO) the first L1A is issued. The BC information is dequeued from the TTC latency FIFO. The subsequent L1As are issued by comparing the counter values with the peak data from this FWFT FIFO. The OCR and the L1As are issued such that the time elapsed since the sending of the corresponding L0 packets is equal to the configured L1 trigger latency (i.e. between 10 and 185 μs in 1 μs steps). This interval is influenced by the set latency, the configured offset and rollover values for the BCID counters and the BC information of the first L0 packet. Some of the selection commands can be masked to test the filtering capabilities of the ROC. The selection ratio can be set from 1 to 100% in steps of 1% and the selection pattern can be configured. The selected BCs are enqueued in the expected FIFOs of the four SROC Capture output data analyzers. The TTC stream is also serialized using the 320 MHz MMCM clock signal and thus can be delayed by half a bit.

Each SROC Capture output data analyzer evaluates the data stream from one SROC. Its architecture is detailed in Figure 3.8. The two serial lines corresponding to the SROC output data are connected to a DDR deserializer. The output of the deserializer is a 10-bit word that is passed to an alignment module. This module searches for *comma* symbols on all 10-bit sub-sequences of consecutive bits within a 20-bit sequence formed by the concatenation of two consecutive deserializer words. The *comma* symbols have unique patterns that cannot be formed by joining any other 8b10b symbol. When two consecutive *comma* symbols are detected, the alignment to the input stream is achieved. The SROC sends a continuous stream of *comma* symbols when it does not have any data to transmit. The alignment is lost if more than 10^6 8b10b symbols are received without any *comma* symbol or if two consecutive *comma* symbols are detected with another alignment position. The aligned 10-bit words are passed to an 8b10b decoder which supplies the decoded data to the *Assembler FSM*. This FSM checks the coherency of the data, the protocol syntax, the parity bits of every hit, the checksum and reported hit words number, the hit content, the BC information and all status bits for each packet. It also forms the corresponding header, hit and trailer words with the same format as the one used in the SROC Packet FIFO (see Figure 2.16). The *Assembler Monitor* is used only in simulations of the test setup RTL in conjunction with the ROC RTL or netlist to record the *Assembler FSM* output data into text files stored on the host computer, for debugging purposes. The resulted files are compared with the output files of the corresponding VMM3 emulators to validate the correct transmission, reception and analysis of the ROC data within the test setup.

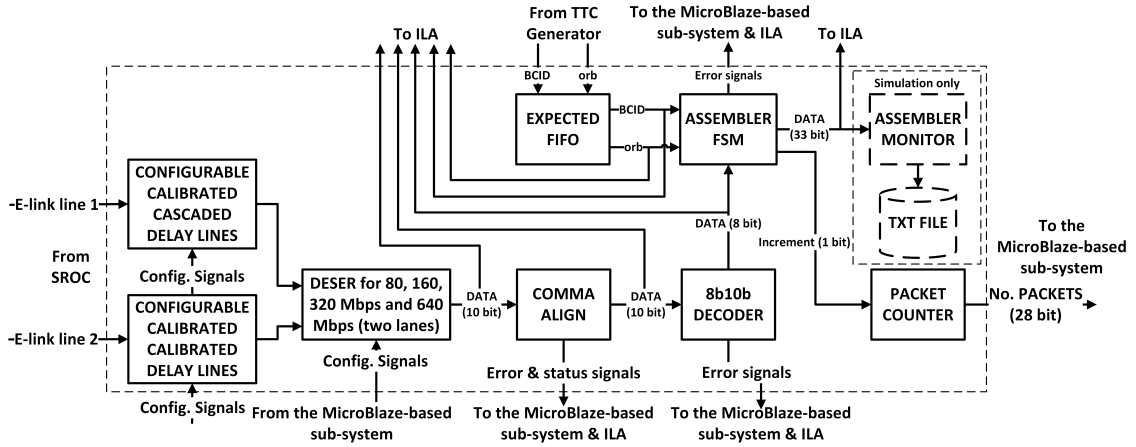


Figure 3.8: The architecture of the output data analyzer.

An ILA [206] with 2^{14} sample space and a configurable triggering system is used to capture predefined signals from all the modules of the *test-bench*. The ILA was a key tool used for debugging purposes and obtaining the results presented in Section 3.2.4 and Chapter 4.

The same 320 MHz clock signal is used for serialization within all the *VMM3 emulators* and the *TTC generator*. As previously described, this clock signal is in phase with the 40 MHz (i.e. BC) and 160 MHz (i.e. RO - optional) clock signals supplied to the ROC. Similarly, the *SROC Capture* analyzers are paced by

the in-phase 160 MHz clock signal. In the NSW context, the ROC supplies distinct and independent phase-adjustable BC and RO clock signals to each VMM3. The VMM3 RO clock signal phase dictates the phase of the incoming VMM3 data. For the L1DDC GBTx to correctly receive the SROC data, the GBTx phase aligners can be used. Alternatively, the phases of the ROC internal clock signals can be adjusted. The phases of the internal ROC clock signals also dictate how the TTC stream is interpreted (see Chapter 2, Section 2.3.1). An increase in complexity is avoided within the functional digital test setup by using aligned clock signals for serialization, deserialization and processing. Still, correct capturing of the serial data must be achieved both in the FPGA and the ROC.

The calibration of the phases for the high-speed serial data streams coming out of and into the FPGA, relative to the capturing clock signals within the ROC and the FPGA is achieved using configurable built-in FPGA delay lines. These delay lines are placed between the serializers and the output pads in the case of the *TTC Generator* and *VMM3 emulators* modules and between the input pads and the de-serializers in the case of the *SROC Capture* monitors. As previously emphasized, all the clock signals within the FPGA test setup are aligned. The BC clock signal (40 MHz) is also forwarded to the ROC's ePLLs which generate its internal and external clock BC and RO clock signals. The phase difference between the output clock signal of the ePLL and the input reference clock is unknown, it depends upon the supplied voltage, the temperature and the chip sample but can be adjusted using its internal delay lines in steps of 200 ps. The ROC internal triplicated RO clock signals must sample the FPGA output serial data as close as possible to the middle of the bit to receive it correctly and avoid meta-stability. Similarly, the FPGA 160 MHz clock signal must sample the ROC output serial data when it is stable. The delay lines specific to the Xilinx Ultrascale architecture were used instead of multiple phase-adjustable clock domains (as in the NSW front-end readout system) due to the resulted smaller complexity, higher resolution (steps of up to 4.16 ps compared to 200 ps in the ePLL case), faster set-up and automatic and continuous voltage and temperature compensation done by IDELAYCTRL primitives built-in the used FPGA device [203]. In addition, this method permits the independent calibration for each channel. The built-in ODELAY and IDELAY primitives [203] are used for FPGA output and input serial data signals, respectively. Both contain a 512-step delay line whose value can be incremented, decremented or loaded with a supplied value. According to [211], the propagation time through one step is in the 2.5 - 15 ps interval. The empirically obtained value is 4.16 ps/step which means that the entire delay line covers only 2.13 ns out of the 3.125 ns bit duration. Except for the TTC stream and the two I^2C interfaces all the other serial data streams between the ROC and the FPGA are 8b10b encoded which means that as long as the data is sampled when stable, the receiver end can infer where each byte begins. The delay lines can be cascaded as described in [203] and illustrated in Figure 3.9 to cover a larger delay interval. For an output data channel, an ODELAY master primitive is cascaded with an IDELAY slave primitive from the unused IOB of a specific neighbor. For an input data channel, an IDELAY master primitive is cascaded with an ODELAY primitive from the same IOB. In this case, the IOB contains internal dedicated routes between its associated delay lines. In case the input signal is differential and the Input Buffer Differential Signaling with Differential Output (IBUFDS DIFF-OUT)

primitive [203] is used, both the negative and the positive paths can be delayed separately, as shown in Figure 3.9, without constraining the other neighbor IOBs. The user can then choose the polarity of the resulted signal as shown in Figure 3.9 using multiplexers. This cascading system is useful for determining the exact value of the delay step as described in [172] and was used for all the differential FPGA test setup's input data channels. For the differential FPGA output data channels cascading was not employed because not all channels were able to borrow delay lines from neighbor IOBs due to the chosen placement. In these cases, a single delay line was used in combination with the presented option of delaying the data stream with half a 320 MHz clock cycle (i.e. 1.5625 ns).

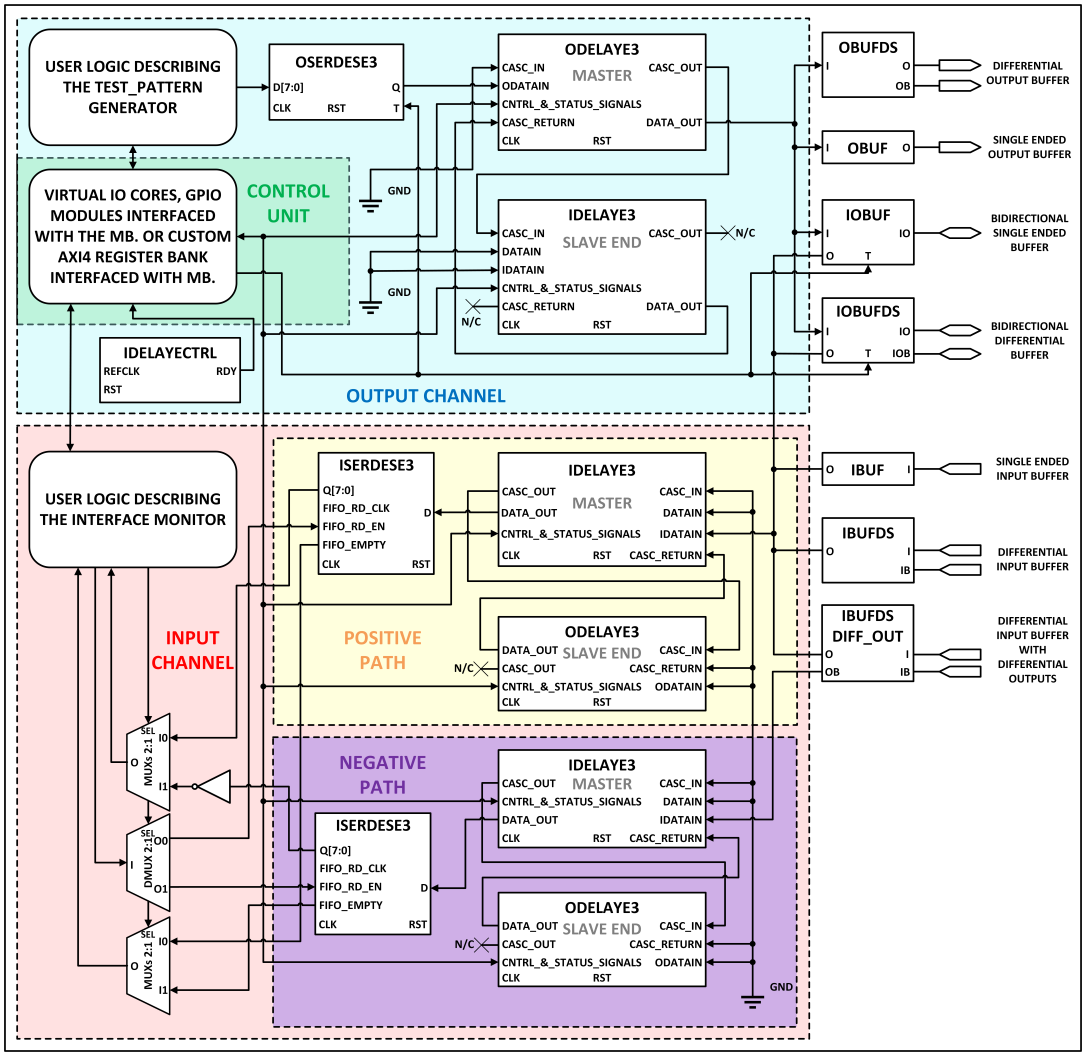


Figure 3.9: The cascading of two delay lines for the synchronization of the input and output serial channels specific to the Xilinx Ultrascale architecture.

MicroBlaze-based testing software

A single-thread standalone C program was designed, implemented, validated and executed on the instantiated MicroBlaze microprocessor to monitor and control the entire ROC test setup. The peripherals from the *MicroBlaze-based sub-system* (i.e. the AXI UART Lite, AXI I^2C, AXI Timer and the custom AXI Register Bank), have

address spaces allocated in the 32-bit virtual memory. The access to the custom AXI Register Bank and the configuration and status registers of the other peripherals is thus made using pointers. Custom Interrupt Service Routine (ISR) functions were designed and used for the timer and the I^2C master. The UART interface is used as standard input (stdin) and output (stdout) channels. A serial terminal application that runs on a host computer acts as the user interface and logs the transferred data into files.

Algorithm 2 The ROC Testing program pseudo-code.

```
1: if init_&_test_MB_periph() then return 1
2: next_chip ← true                                              ▷ global variable
3: while next_chip do
4:     chip_id ← read_chip_id()                                  ▷ global variable
5:     result_calib_out_0 ← calib_fpga_out_ser_data(shift = 0)
6:     result_calib_out_1 ← calib_fpga_out_ser_data(shift = 1)
7:     result_calib_out ← max(result_calib_out_0, result_calib_out_1)   ▷ global variable
8:     result_calib_in ← calib_fpga_in_ser_data()   ▷ global variable; cascaded delay lines
9:     ttc_stream_start ← determine_ttc_stream_start()           ▷ global variable
10:    result_calib_ttc_0 ← calib_ttc_stream(shift = 0)
11:    result_calib_ttc_1 ← calib_ttc_stream(shift = 1)
12:    result_calib_ttc ← max(result_calib_ttc_0, result_calib_ttc_1)   ▷ global variable
13:    if not result_calib_out or not result_calib_in or not result_calib_ttc then
14:        result ← manual_mode() ▷ updates next_chip and returns 1 if the current chip is rejected, 0 otherwise
15:        if result then continue
16:    for test ← 1; test ≤ 10; test ← test + 1 do
17:        result_pu ← power_up_ROC()
18:        result_tb_config ← write_tb_config(test)
19:        result_roc_config ← write_roc_config(test)
20:        if result_pu or result_tb_config or result_roc_config then continue
21:        result ← run_test(test)
22:        result_test ← check_test_result(test)
23:        if (result or result_test) and (test == 1) then
24:            option ← ask_if_continue()  ▷ returns 1 if the user aborts the testing of the sample, 0 otherwise
25:            if option then break
26:    chip_status ← check_all_tests_reports()
27:    if chip_status then print BAD
28:    else print GOOD
29:    next_chip ← ask_if_new_chip()
30: return 0
```

The pseudo-code of the program is depicted in Algorithm 2 and the called functions are detailed in Table 3.2. The functions which do not return specific values return *0* if they complete without errors and non-zero values otherwise. First, the peripherals are initialized and tested. Then, in a loop with one iteration per chip sample, the program receives from the user the unique ID of the sample and executes the automatic delay lines calibration. If the calibration is successful, the program continues with a sequence of ten tests represented as a for loop with one iteration per test. If the automatic calibration fails, an interactive menu of options is presented

Function	**Description**
init_&_test_MB_periph()	Initializes and tests the peripherals from the MicroBlaze-based sub-system.
read_chip_id()	Reads from the user the unique ID of the chip sample.
calib_fpga_out_ser_data(*shift*)	Determines and returns the best delay line settings for all output serial data channels; *shift* enables data stream shift with half a bit.
calib_fpga_in_ser_data()	Determines and returns the best input delay line settings.
determine_ttc_stream_start()	Determines and returns the starting bit of the TTC stream.
calib_ttc_stream(*shift*)	Determines and returns the best delay line settings for the TTC stream; *shift* enables TTC stream shift with half a bit.
manual_mode()	Interactive menu where the user can change the settings of all delay lines, recall the automatic calibration functions, decide if to continue testing the current chip sample, end the program or skip to the next chip sample.
power_up_ROC()	Performs the ROC and test-bench sub-system power-up sequence. Initially enables all the reset signals and then progressively disables them. The settings for all the delay lines are set to the calibrated values.
write_tb_config(*test*)	Writes the test-bench sub-system configuration for the *test*.
write_roc_config(*test*)	Writes the ROC configuration corresponding to the *test*.
run_test(*test*)	Starts the *test* and returns after its run-time elapses.
check_test_result(*test*)	Populates a data structure with the flags, statistics and measurements related to the ROC behavior after the *test*. Prints the *test* report and returns the result.
ask_if_continue()	Determines if the user wants to continue testing the current chip sample.
check_all_tests_reports()	Checks if any test failed, prints a summary and returns.
ask_if_new_chip()	Determines if the user wants to test another chip sample.

Table 3.2: Description for the functions called in the ROC testing Algorithm 2.

to the user. Through this menu, the user can perform manual calibration, recall the automatic calibration functions and decide if to continue testing the current sample, end the program or skip to the next chip sample. Before each test, the ROC is powered up and then configured. The test setup configuration is next applied through the custom AXI Register Bank. The test is started and so is the AXI timer. When the timer expires, its ISR function is called and simply clears a flag. In the main program the timer flag is polled. When the polling loop finishes it means that the test has ended. The test setup and ROC statuses are read, evaluated and printed. After all tests complete a global summary of their results is printed and a final two-choices verdict (*good*/*bad*) is given to the sample. The user is asked if the program should terminate. The test setup does not require resetting or reprogramming for each chip sample.

The *calib_fpga_out_ser_data*, *calib_fpga_in_ser_data* and *calib_ttc_stream* functions sweep the delay lines associated with the *VMM3 emulators*, *SROC Capture* analyzers and *TTC Generator*, respectively. For each phase, the ROC is powered-up

and configured alongside the test setup in an interface-specific scenario aimed at producing as many transitions as possible and then, for 20 ms varied but deterministic valid data are sent to and/or received on the targeted interface. The sweep is repeated with all the data streams shifted by half a 320 MHz clock cycle for the FPGA output channels. For each channel, the delay line phase is set as the middle of the biggest interval of consecutive phases with no errors. An optimized approach [172] that significantly reduces the time of the search is presented in Section 3.2.3. Since the TTC stream is not 8b10b encoded, the byte start position in the stream of bits has to be determined before the delay line calibration. This is achieved by the *determine_ttc_stream_start* function which sends one-hot byte values and observes the effect on the ROC output TTC debug pads.

The automatic calibration works for each good chip and fails for chips that have at least one bad input or output channel. A *manual* mode allows the user to manually set a delay value or to loop multiple times through the calibration functions. The user can also use the ROC ePLLs to shift the RO clock signal relative to the FPGA data.

Ten tests were designed for mass-testing, to cover all functional features of the ROC. Their trigger and throughput parameters are detailed in Table 3.3. All the addresses from all the ROC SRAMs are written with varied but deterministic bit patterns (from the input packets) which are then read and checked for coherency (using the output packets). The packet data is designed so that every bit is toggled multiple times, to detect any fault in the memories. The injected VMM3 packets have either pseudo-randomized or constant length. The packet rate is variable in some tests and constant in others. Different crossbar configurations are tested. Each SROC output channel is tested at all four speeds. The latency of the L1 trigger is varied from 25 to 100 μs. Different settings for the output packets format are checked. The ROC status and control registers are tested. If any bit is incorrect in any field of any output packet for any test the corresponding chip fails.

Test #	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Run Time [s]	5	10	10	10	10	10	10	10	5	10
L1A latency [μs]	100	80	75	55	45	35	25	100	70	35
Average input packet rate [MHz]	1.37	1.363	1.003	1	1	0.2	0.2	1.003	1	1
Average encoded VMM3 data throughput [Mbps]	245.32	152.15	61.16	67.02	67.02	47.04	47.04	61.16	260	220

Table 3.3: Test Bench sub-system settings for the suite of ten tests.

The first test checks the ROC in its default configuration. Test 2 validates the ROC behavior when the VMM3s are sending packets with the following types of set flags and special BC identifiers: (i) hit words with the T bit set; (ii) empty headers with the V bit set and (iii) headers with the BCID value of *PATTERN* and incorrect parity bit (see Algorithm 4 from Appendix A.1). After the latter situation happens, an unknown number of VMM3 packets will be missing from that VMM3 and the ROC must not wait for them or timeout. In this case, it must discard the associated hit data and instead send dummy hits. Tests 3 to 7 are used to validate different

crossbar configurations, output speeds, packet sizes and packet format settings. Test 8 determines the validity of the *bypass mode* in which the received data is passed through ROC without BC information matching. Test 9 applies the worst-case VMM3 packet and BC selection commands burst loop. Test 10 emulates problems on the data paths coming from all the VMM3s causing the ROC to automatically enter into a timeout state after a configurable time. Then the input data streams are re-enabled and the ROC timeout recovery procedure is applied.

3.2.3 Clock and data synchronization method

Reliable synchronous and asynchronous signaling between the testing device and the DUT is essential for ASIC design prototype validation and chip mass-testing. In all data communication channels the receiver end must be able to determine from the input signals the units of data (e.g. bytes) and their correct value. In synchronous communication, a clock signal synchronizes the receiver to the transmitter [149]. This clock signal is provided either by the transmitter to the receiving end (source synchronous interface, e.g. I^2C [157], SPI [154], etc.) or by an external common source (system synchronous interface, e.g. Peripheral Component Interconnect - PCI [122], PCI Express - PCIe [182], etc.) to both ends of the communication channel.

In serial asynchronous communication, the receiver determines the transmitter rate using the transitions of the input data signal (e.g. USB [89]). A locally generated clock signal is adjusted in phase and frequency so that it samples the data when stable and at the correct pace. Alternatively, a handshake circuit [118] [155] [156] can be used for transmitting data without providing a synchronized clock signal.

The relative timing of any signal generated by a device relative to the capturing clock signal within the receiver device is often unknown. In these cases, one can assume that the input signal will not meet the setup and hold requirements of the capturing FF. This causes meta-stability in the receiving device, meaning that the capturing FF is driven into an unstable state. [118] states that the FF will recover from this state in a time interval with an exponential distribution. Even if the meta-stability resolves fast [118], the final stable state might be incorrect. Even if correct, the next FF or register could also enter into meta-stability because the combinational logic between them had insufficient time to settle down. The device could be driven into unforeseen states from which it cannot recover until the reset signal is pulsed.

A correctly constrained and implemented design with passed exhaustive verification through simulation (including gate-level with SDF back-annotation) and STA in all corner cases assures that in a flawless fabricated sample, no meta-stability will occur as long as the supplied input signals satisfy the imposed constraints and the environment conditions (supplied voltage, nuclear and electromagnetic radiation, temperature) are kept within the limits imposed by the manufacturing technology. The input and output constraints for the device pins are often difficult to estimate a priori because they imply knowing various characteristics of the exterior like the propagation time from the generating device pad and the parasitic resistance and capacitance of this path affecting the rise and fall times of the signal. Furthermore, these characteristics are dependent on various parameters, mainly the temperature and the supplied voltage [149]. Consequently, the ASIC designers often use phase-adjustable capturing clock signals and/or configurable delay lines on the data path.

When trying to assess the usability and validity of a device sample it is important to correctly time the input and output signals to and from the DUT to correctly differentiate between a *failed* and a *passed* sample. The only cause of failure should be reduced to a fabrication defect in the DUT. In the ROC digital functional testing case, two synchronization mechanisms for the multiple input and output independent source-synchronous communication links were implemented and compared: (i) a basic sweep on data delay lines and (ii) a more efficient delay line scanning based on a double binary search.

The two ROC internal TMR clock domains (i.e. 40 MHz BC and 160 MHz RO) are mesochronous to the 40 MHz and 160 MHz clock signals used and supplied by the FPGA. Two clock signals are mesochronous if they have the same frequency but an unknown stable phase difference between them.

Table 3.4 lists the main data signals of the FPGA-ASIC system. The ASIC contains 16 DDR 320 Mbps source-synchronous serial input links. The links are aggregated together in groups of two forming eight 640 Mbps input data lanes (i.e. VMM Capture serial input - see Chapter 2). They transport L0 data with 8b10b encoding. This line encoding is mainly used for determining the position of the data symbols (10-bit wide) within the data stream regardless of when the capturing process begins. Four 640 Mbps 8b10b encoded output data lanes, each formed similarly through the aggregation of two DDR 320 Mbps links, forward the DUT's processed data back to the FPGA. In both cases, as long as the data lines are sampled within the open region of the eye diagram the DUT or the FPGA can infer where each data word begins.

Data signal name	8b10b	Description
FPGA_TX_8B10B_DATA	Yes	encoded output from the FPGA (launched L0 data)
DUT_RX_8B10B_DATA	Yes	encoded input to the ASIC (arrived L0 data)
FPGA_TX_DATA	No	output from the FPGA (launched TTC stream)
DUT_RX_DATA	No	input data to the ASIC (arrived TTC stream)
DUT_TX_8B10B_DATA	Yes	encoded output from the ASIC (launched L1 data)
FPGA_RX_8B101B_DATA	Yes	encoded input to the FPGA (arrived L1 data)

Table 3.4: The serial data signals within the FPGA-ASIC system.

Another DDR 320 Mbps source-synchronous serial DUT input channel captures uncoded control data (i.e. the TTC stream). The beginning of the data unit (byte) is determined using the positive edge of the internal PLL generated 40 MHz clock signal. The phase difference introduced by the DUT's PLL relative to the mesochronous reference can be larger than the duration of one bit. Since the data is not 8b10b encoded there is no direct method of determining where each byte begins and as a result, data can be misinterpreted. Consequently, a control command is issued on the stream, one bit position at a time and some debug DUT pins are monitored. If the expected effect happens then the correct alignment has been found.

In Figure 3.10 the timing dependencies between data and clock signals in the FPGA-ASIC system are depicted. For simplification, the 8b10b encoded data lanes are DDR 320 Mbps rather than 640 Mbps with the bits interleaved on two links since in both cases the alignment can be determined similarly. As in Chapter 2, Section

2.3.1, the bits are tagged with one letter representing their position within the data unit and one digit referring to the data unit count. The unknown propagation delay between FPGA and ASIC and the unknown clock signal phase shift introduced by the ROC's ePLL are depicted as *prop. delay* and DUT_PLL_phase_delta, respectively. Due to these delays, the first bit of the captured byte on the TTC stream will be B instead of A in the illustrated case. The depicted clock signals are detailed in Table 3.5.

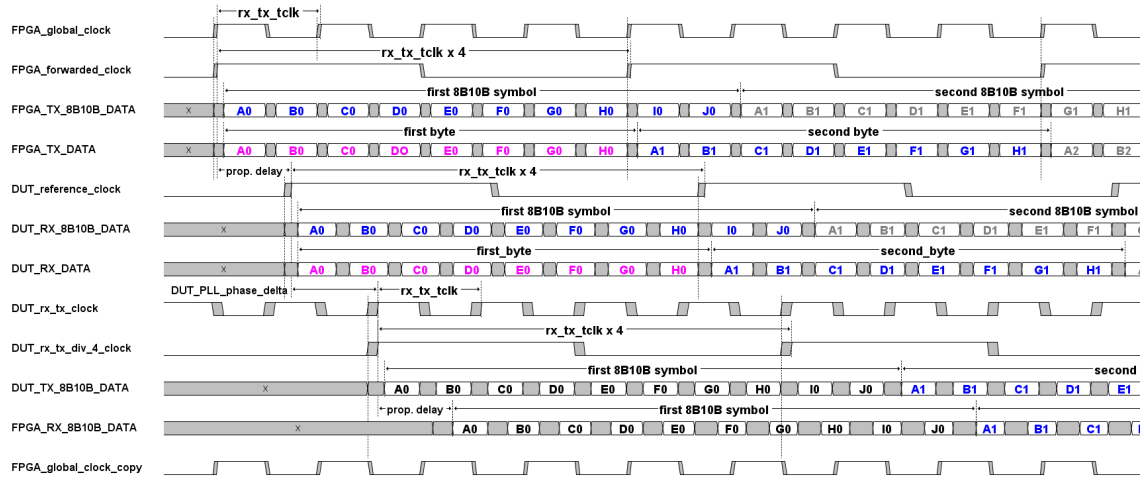


Figure 3.10: Simplified waveforms depicting the timing between the DDR serial data lines and the clock signals within the ROC digital functional testing system.

Clock signal name	Frequency	Description
FPGA_global_clock	160 MHz	used by the FPGA logic for processing and for serial transmission and reception (RO).
FPGA_forwarded_clock	40 MHz	BC clock forwarded to the DUT, in phase with FPGA_global_clock.
DUT_reference_clock	40 MHz	DUT's reference BC clock for its PLLs, represents the propagated and received FPGA_global_clock.
DUT_rx_tx_clock	160 MHz	used in DUT for serial transmission and reception (RO) and data processing, generated by its PLL from DUT_reference_clock.
DUT_rx_tx_div_4_clock	40 MHz	used in DUT for processing, generated by its PLL from DUT_reference_clock.

Table 3.5: Descriptions of the clock signals from the ROC digital functional testing system that appear in Figure 3.10.

Three methods of adjusting the phases between the clock and serial IO data signals were identified: (i) adjusting the phase of the reference BC clock signal supplied by the FPGA, (ii) adjusting the phases of the ePLL output clock signals and (iii) shifting the serial data independently on each IO channel using delay lines built-in the FPGA. The last option was chosen because it allows an independent and more accurate calibration for each channel.

In a first approach, the automatic synchronization mechanism has been developed as a software component, running on the instantiated MicroBlaze micropro-

cessor from the FPGA digital functional test setup, that sweeps the delay lines and validates the responses from the ASIC and FPGA. The middle of the largest interval of consecutive valid delay steps is the chosen delay value. At each step, the injected valid data is varied as much as possible to decrease the MTBF as defined in [118]. For the 8b10b channels, the status flags of the receiving end are checked for decoding, parity and coherency errors. If no error occurred the delay is valid. The last channel that is calibrated is the TTC stream. The expected behavior is checked considering all the other data channels already calibrated in this case.

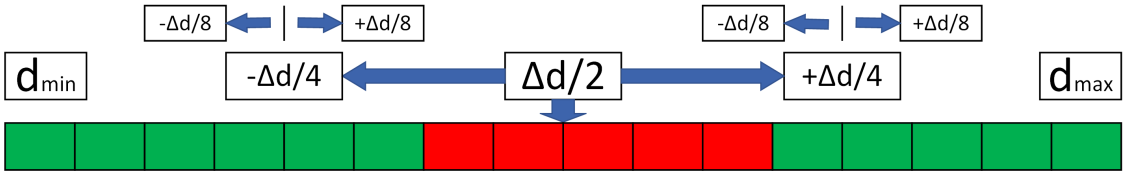


Figure 3.11: The implemented and used double binary search calibration mechanism.

Because checking all possible delay values is inefficient, a double symmetrical binary search inspired by the classical binary search algorithm was implemented. The proposed approach is depicted in Figure 3.11 where the valid delay steps are represented with green and the invalid ones with red. The presented scenario is worst-case because the algorithm must determine the largest of two valid intervals. The delay line extremities are d_{min} and d_{max}. The algorithm first samples the middle of the delay interval ($\Delta d/2$), where $\Delta d = d_{max} - d_{min}$, and then moves in both directions with $\Delta d/4$. Then, from each of the two resulting positions it moves with $\Delta d/8$ in one or both directions based on the previous results and so on. Until the desired resolution is reached, the step size is halved at each iteration. The complexity of the proposed algorithm is $O(logN)$ compared to $O(N)$ for the classical sweep.

3.2.4 Experimental testing results

The RTL and netlist codes of the ROC's packet processing logic and the digital functional test setup RTL were validated in simulations in which they were interfaced. The implemented FPGA design met the timing, placing and routing constraints. The first real-world ROC chips were extensively tested in different scenarios for periods ranging from a couple of minutes up to 48 hours and ILA-captured data were analyzed. Further on, the ROC passed preliminary integration tests with the other ASICs from the NSW context [189] [190] [151] [61]. The faster calibration method was successful. The functional digital test setup was used for chip mass-testing. In this section the digital design validation coverage is presented, the theoretical model that describes the chip performance depending on its configuration and the input packets characteristics is validated, the chip's behavior related to data loss when the rates are above the theoretical thresholds is detailed, the results of the faster calibration method are presented and the mass testing results from Transilvania University of Brașov are analyzed.

Digital design validation coverage

This sub-section presents the functional features of the ROC packet processing logic that were validated using the presented functional digital test setup on the ASIC prototypes. Together with the features covered by the analog test setup, they assured the design and implementation correctness and the quality control of the manufactured chips.

During the ROC's digital design validation, it was checked that all the receiving channels from the ROC and the FPGA design reach and maintain the alignment state when only *commas* are sent, for all transmission speeds. Then, it was confirmed that this also happens when packet data are sent. Also the correct deserialization, decoding, checking and buffering of valid VMM3 packets of various sizes within all VMM Capture channels was confirmed. It was confirmed that the VMM3 packets with damaged header are discarded. The correct deserialization, interpretation and buffering of TTC selection and synchronization commands and the correct buffering, encoding and transmission of output data packets were validated. The correct forwarding of the TTC commands using both logic analyzers and oscilloscopes was observed. The ROC digital part register bank was accessed through I^2C and the correct configuration and reporting of errors and status flags were validated.

Once all the ROC interfaces were validated, the processing logic was examined. The correct processing of valid input packets with various sizes, rates and content by matching the TTC selection commands was confirmed. The unselected input data was correctly dismissed. The ROC maintained the synchronicity between the selection commands and the input data. The *bypass mode* was successfully used. The correct data routing through the crossbar in different configurations, the correct suppression of the TDC field from the hit words and the correct suppression of the output null-events were confirmed. The correct back-to-back output packet transmission with or without the suppression of the EOP symbol was validated. The correctness of the communication protocol, the syntax and the expected values for all the output packets were confirmed. The implemented flow and congestion control mechanisms were triggered and validated. The correct automatic disabling of inactive input channels was confirmed and the recovery procedure after they again become active was successfully applied. The ability to manually disable and then re-enable the input and output channels was confirmed.

Finally, the performance of the ROC as theorized in Chapter 2, Section 2.4 in terms of throughput and maximum data rate without loss, was assessed. The worst-case selection commands and input packets burst loop (see Chapter 2, Section 2.1) was applied. No data loss was observed as long the data throughput is not saturating the output bandwidth (i.e. the theoretical threshold limit is respected).

Maximum input and output packet rates and data loss

The aim was the validation in real-world of the VMM3 and SROC maximum transmission rates (i.e. $\lambda_{VMM3\_max\_tx}$ and $\mu_{SROC\_max\_tx}$) deduced in Chapter 2, Subsection 2.4.3 based upon the interfaces and protocols. The assumptions are: (i) the VMM3 and ROC ASICs can saturate their output interfaces; (ii) each VMM3 produces the same number of hit words for each L0 trigger (i.e. $n = \bar{n}$), (iii) all the VMM3 packets are selected in the ROC (i.e. $sel = 100\%$); (iii) no data loss occurs within VMM3; (iv) the latency of the L1A selecting commands does not cause the overflow

of the VMM Capture FIFOs and (v) there are no empty (without hit words) VMM3 packets (i.e. $p = 0$). The equations become[1]:

$$\lambda_{VMM3\_max\_tx} = \frac{1.6 \cdot 10^7}{(1-p)\cdot(\bar{n}+1)+p} = \frac{1.6 \cdot 10^7}{\bar{n}+1}\ [pkt/s]$$

$$\mu_{SROC\_max\_tx} =$$

$$= \frac{k \cdot v}{10[k\sum_{i=1}^{m}\binom{m}{i}\cdot(1-p)^i \cdot p^{m-i}\cdot(9+e+i\cdot\bar{n}\cdot h)\ +k\cdot p^m\cdot(3+e)\cdot o+3-e]} =$$

$$= \frac{k \cdot v}{10[k\cdot\bar{n}\cdot m\cdot h+(k-1)\cdot e+9\cdot k+3]}\ [pkt/s]$$

In Figure 3.12 these theoretical maximum rates of packets are plotted as functions of the average number of hit words $\bar{n}$ in the VMM3 packets for several ROC configurations. Even if only one VMM3 is associated with the SROC, the overhead of the ROC output format limits, in most cases, the maximum rate of packets. As previously stated, the considered maximum rate of VMM3 packets within NSW was 1 MHz during the ROC development. Thus, the ROC performance at 1 MHz was checked with several measurements which are marked in Figure 3.12. They validate both the mathematical models and the ROC and digital functional test setup implementations.

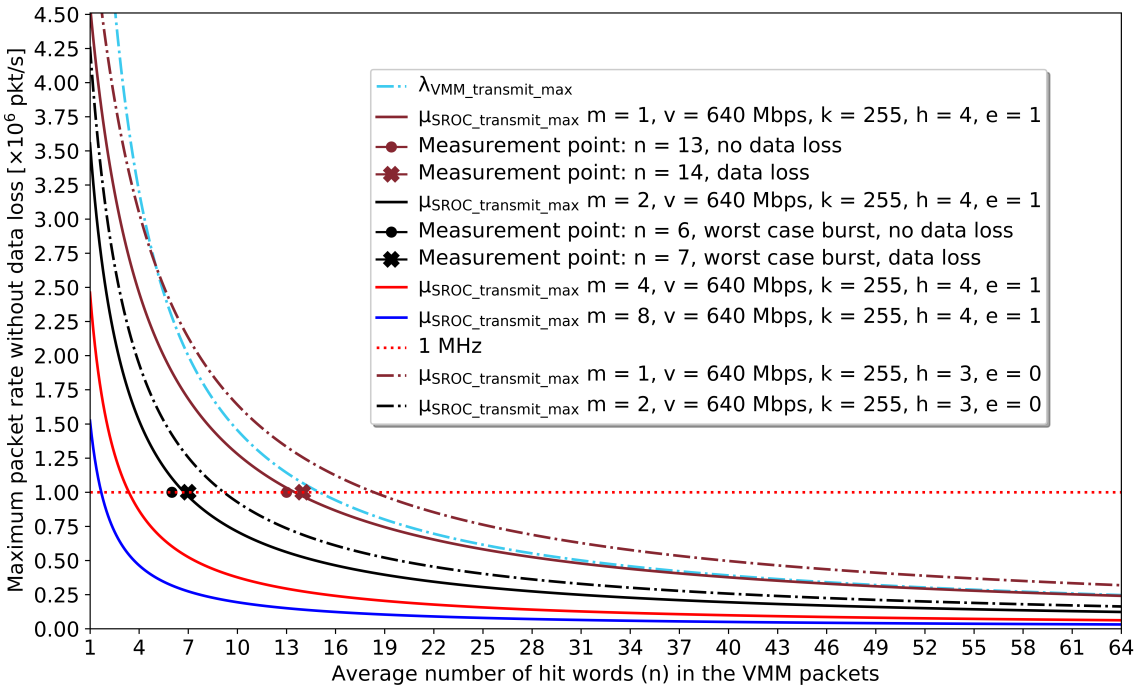


Figure 3.12: Maximum lossless packet rate vs. the average number of L0 hit words.

In Figure 3.13, the SROC effective throughput is traced as a function of the server (i.e. the SROC's Packet Builder FSM) utilization as defined in Chapter 2, Section 2.4, considering the VMM3 packets as the customers. The effective throughput is considered the amount of uncoded packet-only data (no *comma*, SOP, EOP, *Busy-On* or *Busy-Off* control symbols). As *sel* decreases (i.e. the L1 trigger rate is

[1]It was considered that $0^0 = 1$ [66].

lowered), the throughput decreases since fewer output packets are transmitted while the server utilization increases since it must process more VMM3 packets for each L1 trigger. It is considered that the VMM Capture FIFOs never go empty in these scenarios. If the rates are above the theoretical maximum ones, the output interface is saturated while data loss occurs.

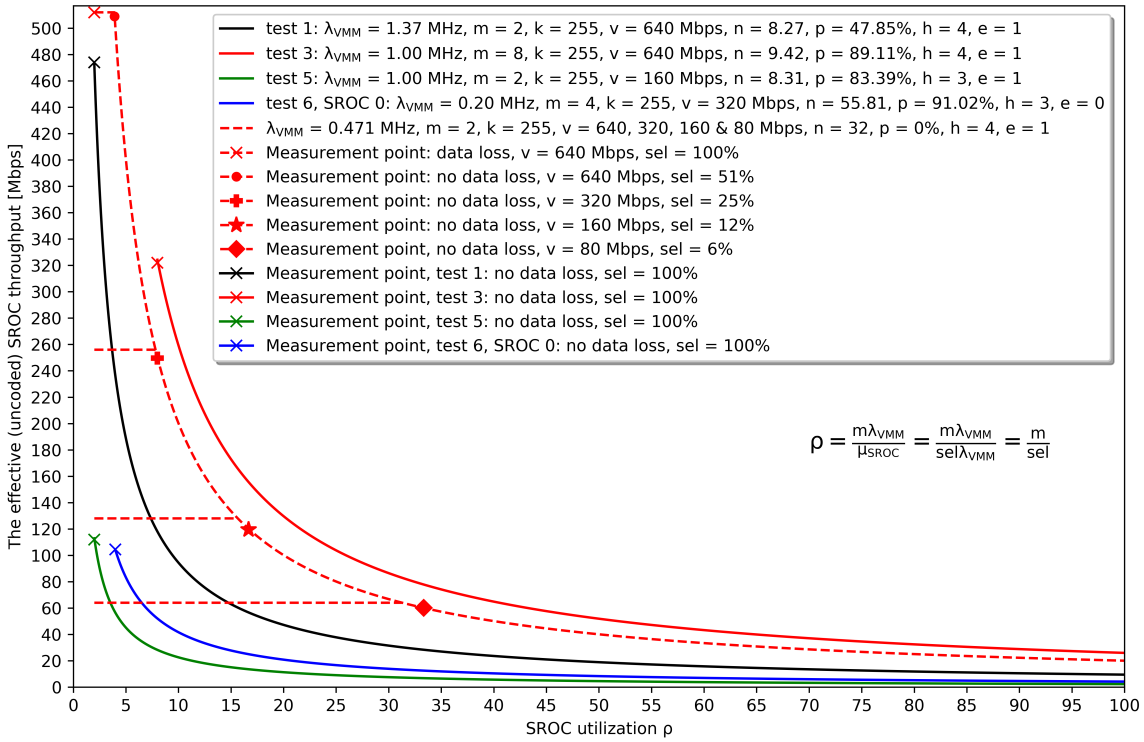


Figure 3.13: The effective (uncoded) SROC throughput as a function of server (i.e. the SROC's Packet Builder FSM) utilization as defined in Chapter 2, Section 2.4.

In Figure 3.14 the evolution in time of the ROC FIFOs occupancy is shown, considering the worst-case packet burst loop (see Chapter 2, Section 2.1) in the conditions marked by the black dot marker from Figure 3.12: $m = 2$, $v = 640 \times 10^6$, $k = 255$, $h = 4$, $e = 1$, $n = 6$. Only one trace for each type of ROC FIFO is depicted for clarity, as the fill levels are similar. Using the equation for $\mu_{SROC\_max\_tx}$, the maximum theoretical SROC packet rate in these conditions is 1.1033 MHz so no data loss should happen since the average packet rate for the worst burst is 1 MHz. The actual fill levels confirm the theoretical expectation. They show periodicity and never reach the maximum buffer size meaning that no FIFO becomes full and data are not lost, as observed at the outputs. By increasing n to 7, the maximum theoretical SROC packet rate decreases to 0.96958 MHz which is below the input average packet rate, meaning data will be lost. The evolution in time of the ROC FIFOs occupancy for this case is depicted in Figure 3.15. The SROC processes the data faster than it can be transmitted causing the SROC FIFO to become full and the SROC processing logic to be throttled. This, in turn, causes the VMM Capture and TTC FIFOs to fill up. The TTC FIFO becomes full first and some BC selection commands are dropped. The rate of packets processed by the SROC is thus limited to the maximum possible to be transmitted. The VMM Capture FIFO never becomes full, so no packet will be truncated. This is possible because the SROC logic can flush the unselected VMM3 packets independent of the SROC

FIFO state. The measured rate of ROC packets equals the maximum theoretical one. A further increase in size for the VMM3 packets could also cause the VMM Capture FIFOs to become full which in turn will cause packets to be truncated or dropped even before they reach the SROC. The selection commands dropped due to a full TTC FIFO can be deduced from the headers or null events words. The entirely dropped or truncated VMM3 packets are signaled identically in the trailer word but can be distinguished by the presence of hit words from that VMM Capture.

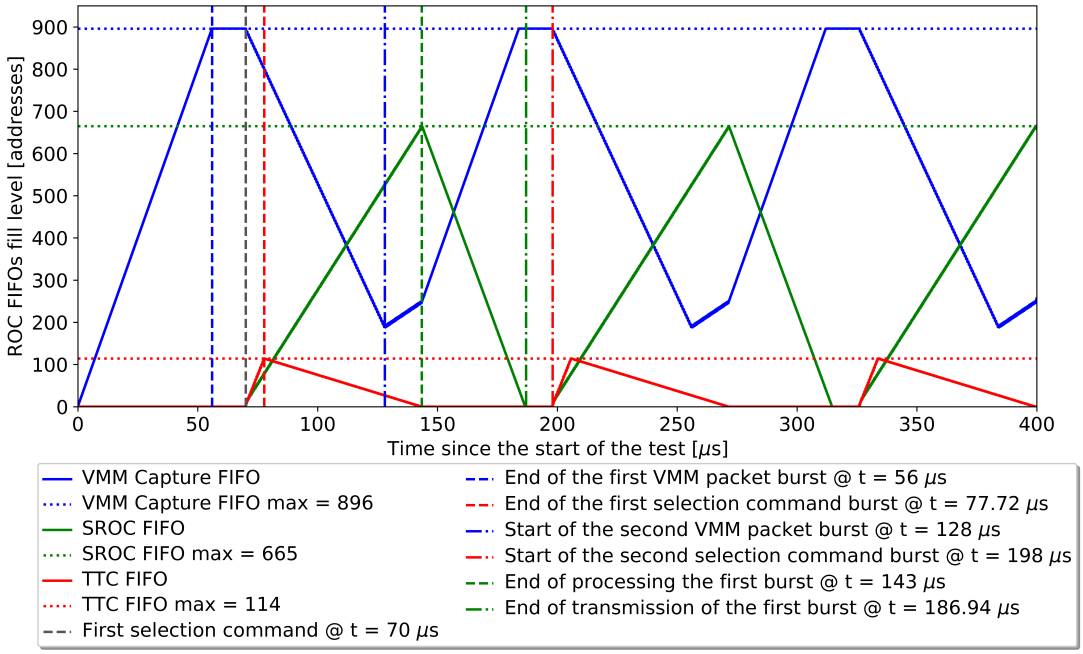


Figure 3.14: ROC FIFOs occupancy for the worst-case packet burst loop (black dot marker in Figure 3.12), showing that no FIFO overflows (no loss).

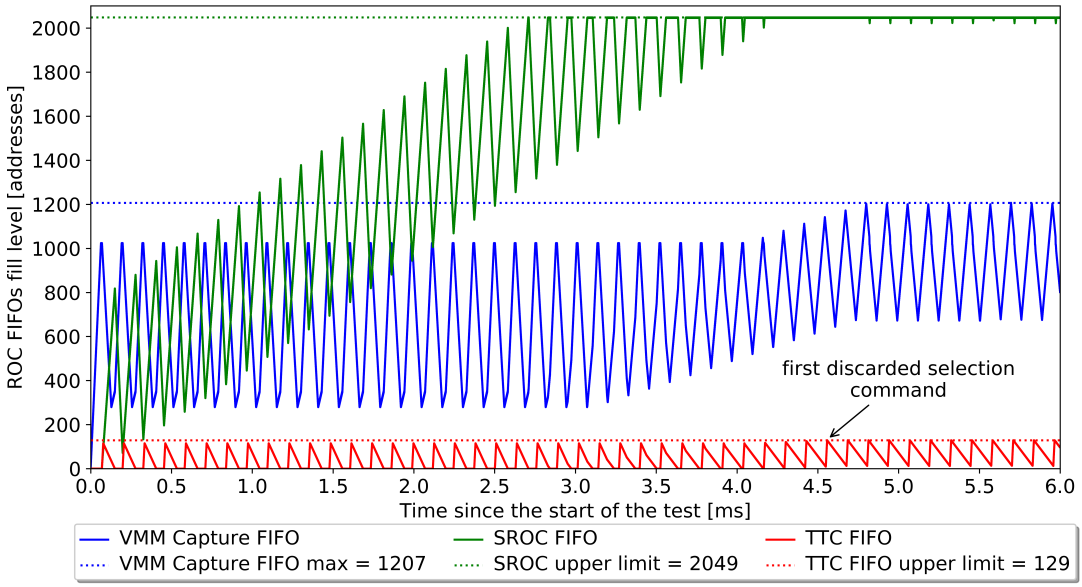


Figure 3.15: ROC FIFOs occupancy for the worst-case packet burst loop (black x marker in Figure 3.12), showing that SROC and TTC FIFOs become full and selection commands are lost.

The ROC requirements state the expected maximum data rates out of VMM3 and ROC ASICs based upon the expected background rates, the type of detectors, their location and area. At $\lambda_{VMM} = 1$ MHz the maximum data rates, including the 8b10b encoding, are 145 Mbps and 285 Mbps for a VMM3 reading MM and sTGC detectors, respectively. Assuming that $m = 2$ and a 100% BC selection rate within the ROC, the associated stated SROC rates are: 256 Mbps and 371 Mbps, respectively. The VMM3 values translate to $\bar{n}$ being 2.625 and 6.5, respectively. Considering m as unknown in the $\mu_{SROC\_max\_tx}$ equation, one SROC with $v = 640 \times 10^6$ can aggregate data from up to 4 such MM VMM3s, up to 2 MM VMM3 at $v = 320 \times 10^6$ or only from one MM VMM3 at $v = 160 \times 10^6$. Similarly, one SROC with $v = 640 \times 10^6$ can aggregate data from a maximum of two sTGC VMM3, regardless of the h value. In Figure 3.12, the continuous black line intersects the dotted horizontal red 1 MHz line at $\bar{n} = 6.749$. Therefore, the ROC can support these anticipated maximum data rates without saturating the output transmission lines even considering the worst-case burst scenario. Also, it can support input packet and BC selection rates above 1 MHz as long as they are below $\mu_{SROC\_max\_tx}$ (e.g. for the same scenario from Figure 3.12 marked with a continuous black line, if $\bar{n}$ is below 2.74 the ROC can process VMM3 packets with 100% selection ratio at 2 MHz without any data loss).

Calibration results

In Figure 3.16 the normalized histograms of the size of the largest valid delay interval (left) and the found optimal delay value (i.e. middle of the largest valid interval - right) for two 8b10b encoded DUT input channels (i.e. VMM Capture 1 and 7), computed over a subset of 347 tested chips supplied with nominal (top) and sub-nominal (bottom) voltages are depicted. The difference between the average sizes for the two channels at nominal voltage is 137.55 delay steps (i.e. 572 ps) despite the identical time constraints and RTL code used in the ASIC design phase. Channel 1 invalid interval is shifted with this amount towards the d_{max} extremity of the delay line resulting in the lower sized valid interval. The spreading of the histograms at 1.1 V is caused mainly by the increased variability of the clock phase shift introduced by the DUT PLLs (DUT_PLL_phase_delta) at lower than nominal voltages. Also, the slew rate of the signals decreases, limiting the area in which the data can be correctly interpreted. As a reminder, the FPGA output channels do not implement cascaded delay lines due to device constraints but instead have the option of delaying the data stream with 1.5625 ns (i.e. half a 320 MHz clock cycle). The presented histograms show the results without this optional shift.

The presented synchronization methods proved to be useful for evaluating the size of the crossing region of the eye diagram (the jitter estimated size) for the DUT input channels. In Figure 3.17 the oscilloscope-measured eye diagrams on a sample chip for an input (top row) and an output DUT channel (bottom row), both at nominal (right side) and at sub-nominal (left side) voltages are shown. The estimated jitter values resulted from the waveforms are 670, 834, 1477 and 1664 ps from top to bottom and left to right showing degradation when lowering the supplied voltage and passing through the DUT, as expected. For the same chip and channels, the sizes of the invalid delay interval are 665.6, 844.5, 561.1 and 773.76 ps in the same order. In this case, the jitter estimates for the output channel are lower than those for the input one, contrary to what was measured in the diagrams, due

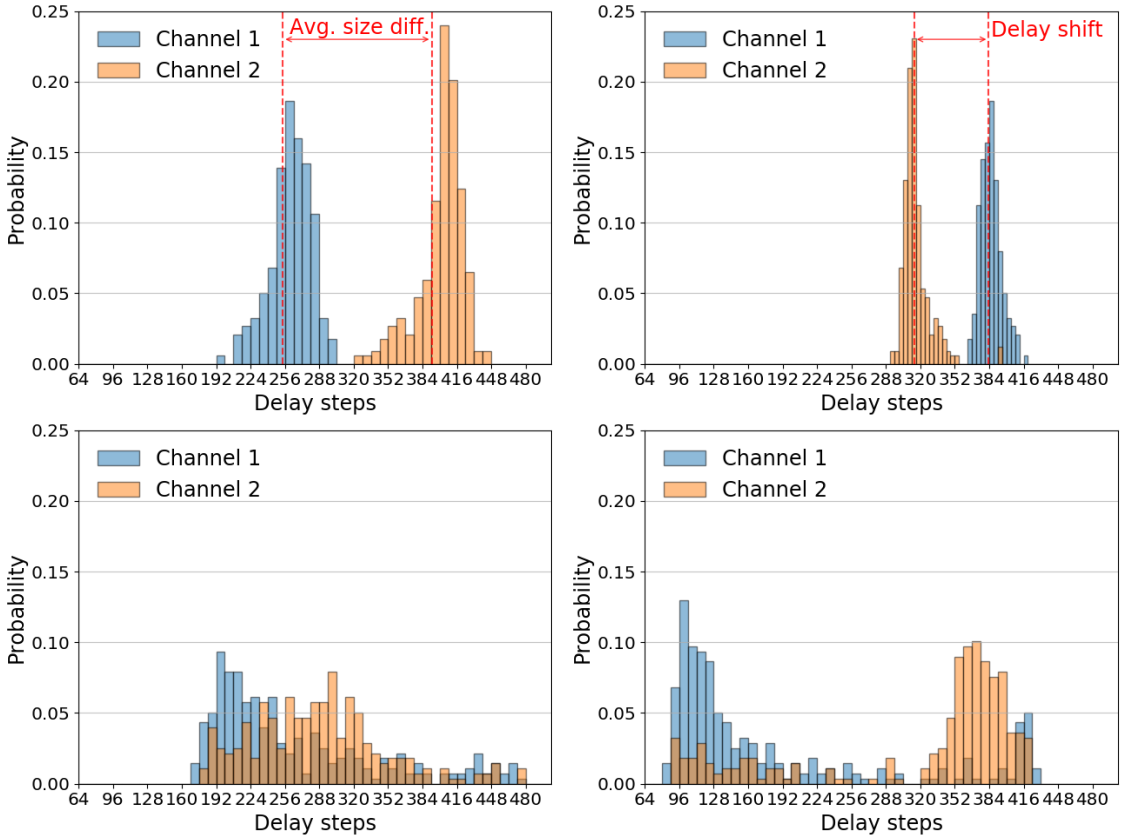


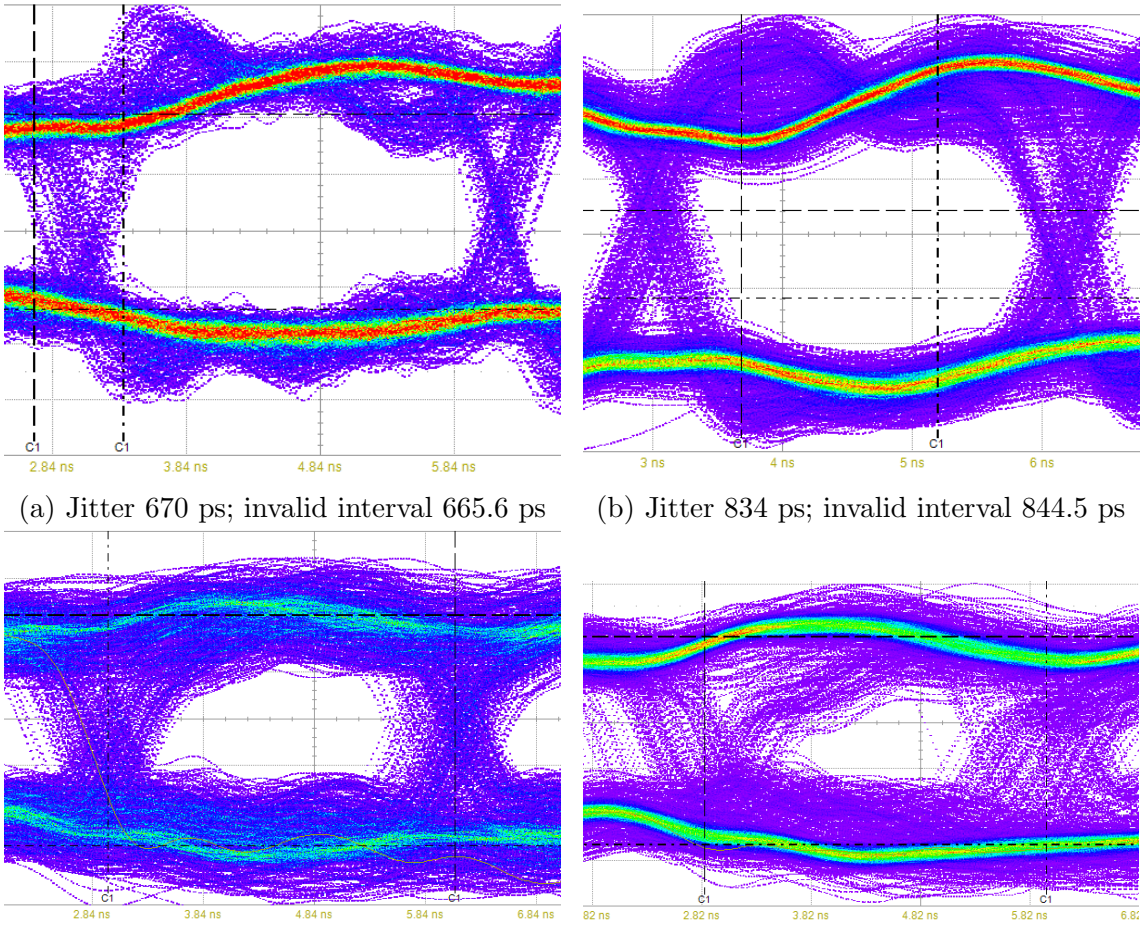
Figure 3.16: For two DUT input channels: histograms for 347 chips of the valid delay interval size (left) and the found optimal delay value (right) at nominal (top) and sub-nominal (bottom) voltages [172].

to the shorter observation duration of 20 ms/step relative to approx. 10 s. For a more realistic value, the observation time must be increased.

The chip testing confirmed the stated computational complexity for both synchronization methods and the improvement introduced by using the proposed double symmetrical binary search. Considering the 512 steps delay line and the desired resolution of 8 steps, the sweeping method constantly finishes after 1.28 s while the proposed double symmetrical binary search ends approx. 70% faster in the worst case (i.e. 0.38 s).

Digital part mass testing

At Transilvania University of Brașov, 2,677 BGA ROC samples were functionally tested both digital and analog. The following descriptions and results refer solely to the digital testing. The 2,677 chips were organized in 4 batches and were tested in two scenarios: (i) supplied with the nominal 1.2 V and (ii) the sub-nominal 1.1 V (for approx. 8.3% drop) on both the core and pads power lines. Only the chips that passed all the tests were marked as *good*. The reason behind the sub-nominal voltage testing is to assure the good performance of the chips in the long run. This sub-nominal voltage was chosen because the used configurable LDO regulators can change the supplied voltage in steps of 50 mV and the intermediate voltage of 1.15 V is below the desired 5% voltage drop threshold. Thus, the tests were performed in worse conditions. As mentioned in Chapter 2, during the ROC digital design



(a) Jitter 670 ps; invalid interval 665.6 ps

(b) Jitter 834 ps; invalid interval 844.5 ps

(c) Jitter 1477 ps; invalid interval 561.1 ps

(d) Jitter 1664 ps; invalid interval 773.76 ps

Figure 3.17: Measured eye diagrams for a DUT input channel at nominal (top-left) and sub-nominal (top-right) voltages and for a DUT output channel at nominal (bottom-left) and sub-nominal (bottom-right) voltages.

implementation, the netlists resulted from the synthesis, placement, routing and sign-off stages were validated in gate-level simulations with back-annotated SDF files in the three corner-cases: typical (i.e. 1.2 V, 25 °C), fast (i.e. 1.6 V, −55 °C) and slow (i.e. 1.1 V, 125 °C). Therefore the design should function properly at the chosen sub-nominal voltage. As a quote, during the design validation testing, the package reached a maximum temperature of 41.8 °C in a 24 °C environment. An example of the test setup's thermal regime of operation with emphasized hotspots is depicted in Figure 3.18. A BGA ROC chip is mounted on a testing PCB (i.e. version 3, depicted in Figure 3.5c, rotated 90° clockwise). The PCB is interfaced with the FPGA evaluation board which continuously emulates the NSW context in the configuration used for the first mass-test while the air temperature is 26 °C. In this case, the LDO voltage regulator supplying the core logic reaches a higher temperature (i.e. 43.8 °C) than the actual ROC chip (i.e. 41.3 °C).

In Table 3.6 the obtained yields per batch and in total are detailed. In Tables 3.7 and 3.8 the distributions of the main causes of failure are detailed for the nominal and sub-nominal voltages, respectively. In Table 3.9 the distribution of the main causes of failure for the chips that passed at nominal voltage but failed the sub-nominal testing is depicted. A failed power-up is caused by an error in the

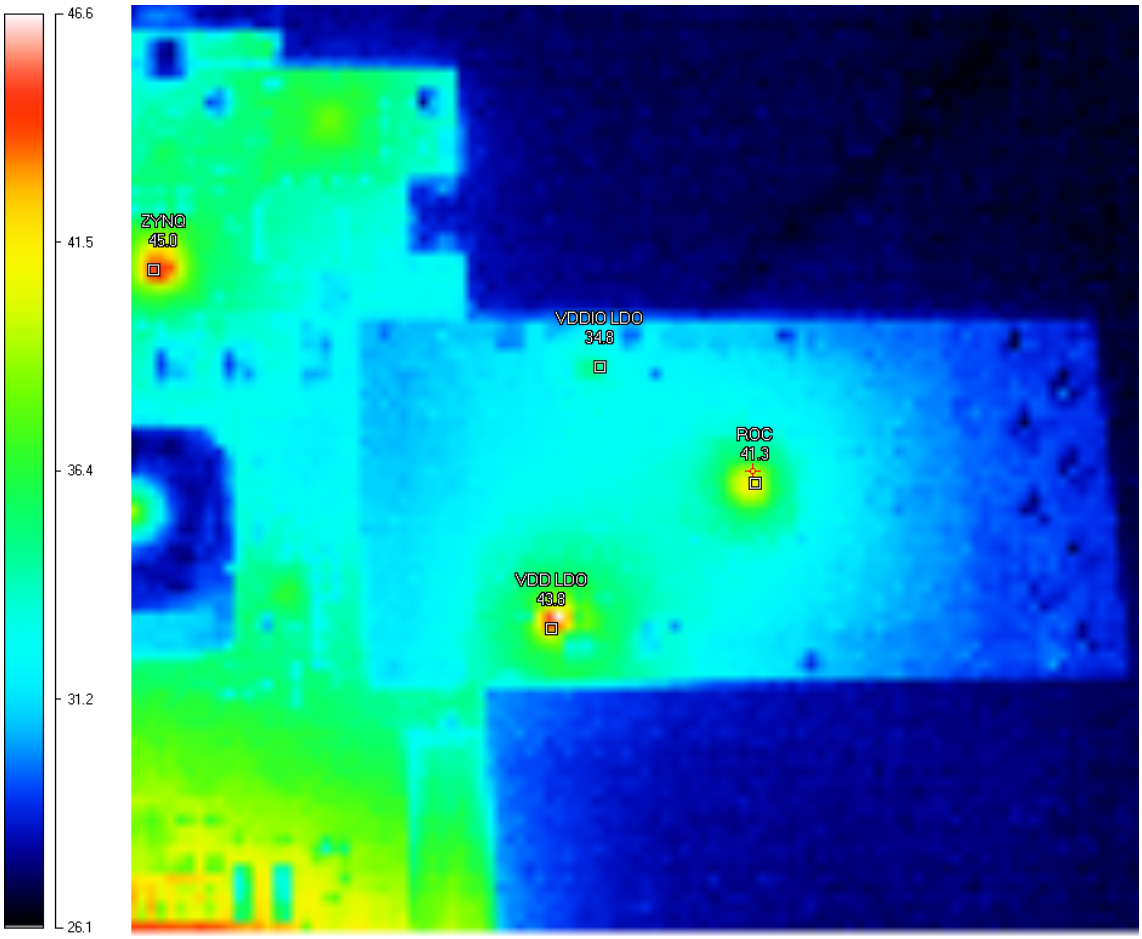


Figure 3.18: Operational thermal regime during digital design validation.

power_up_ROC or *write_roc_config* functions from Algorithm 2 (i.e. the ROC ePLLs do not lock or the ROC I^2C slaves do not acknowledge all the transfers). The failed VMM, SROC and TTC calibrations refer to problems in the *calib_fpga_out_ser_data*, *calib_fpga_in_ser_data*, *determine_ttc_stream_start* and *calib_ttc_stream* functions (i.e. no valid configuration found for at least one channel delay line). A chip that belongs to a listed category will always fail at the categories to the right: e.g. if a chip fails the power-up it will also fail the VMM, SROC and TTC data lines calibrations and the ten tests; if another chip first fails the SROC calibration it will also fail the TTC calibration and the ten tests. Therefore, in Tables 3.7, 3.8 and 3.9 the distributions of failed chips are based upon the first reason of failure during the execution of Algorithm 2. As expected, no chip failed at nominal voltage but passed the sub-nominal testing. The number of failed chips from each category was expected to increase when testing at the sub-nominal voltage compared to the nominal testing. This is not the case for the chips from batch 3 that failed the VMM calibration. Some of the chips that failed the calibration at the nominal voltage failed the power-up sequence during the sub-nominal testing, enough to decrease the count for the sub-nominal testing. This worsening of the root cause of failure happens for chip samples from all categories. A fabrication defect that does not immediately affect the operation of a chip sample in nominal conditions has an increased probability of causing errors when the chip operates in sub-nominal conditions. The ROC digital mass-testing aimed to separate the chip samples that can operate for longer in the LHC environ-

ment so only the chips that passed the digital tests at both supply voltages and the additional test of the analog part will be used within the NSW TDAQ system.

Batch number	**Number of chips**	**Good chips at 1.2V**	**Yield at 1.2V**	**Good chips at 1.1V**	**Yield at 1.1V**
1	337	299	88.72%	206	61.13%
2	450	388	86.22%	329	73.11%
3	810	720	88.89%	566	69.88%
4	1,080	936	86.67%	718	66.48%
Total	2,677	2,343	87.52%	1,819	67.95%

Table 3.6: The digital mass testing results at nominal and sub-nominal voltages.

Batch number	**Failed chips at 1.2V**	**Failed power-up**	**Failed VMM calibration**	**Failed SROC calibration**	**Failed TTC calibration**	**Failed at least one test**
1	38	4 (10.53%)	6 (15.79%)	9 (23.68%)	16 (42.11%)	3 (7.89%)
2	62	12 (19.35%)	14 (22.58%)	15 (24.19%)	10 (16.13%)	11 (17.74%)
3	90	9 (10.00%)	15 (16.67%)	19 (21.11%)	31 (34.44%)	16 (17.78%)
4	144	24 (16.67%)	25 (17.36%)	33 (22.92%)	44 (30.56%)	18 (12.50%)
Total	334	49 (14.67%)	60 (17.96%)	76 (22.75%)	101 (30.24%)	48 (14.37%)

Table 3.7: The distribution of causes of failure at nominal voltage.

Batch number	**Failed chips at 1.1V**	**Failed power-up**	**Failed VMM calibration**	**Failed SROC calibration**	**Failed TTC calibration**	**Failed at least one test**
1	131	16 (12.21%)	7 (5.34%)	18 (13.74%)	30 (22.90%)	60 (45.80%)
2	121	26 (21.49%)	18 (14.88%)	19 (15.70%)	13 (10.74%)	45 (37.19%)
3	244	29 (11.89%)	11 (4.51%)	28 (11.48%)	40 (16.39%)	136 (55.74%)
4	362	48 (13.26%)	29 (8.01%)	44 (12.15%)	59 (16.30%)	182 (50.28%)
Total	858	119 (13.87%)	65 (7.58%)	109 (12.70%)	142 (16.55%)	423 (49.30%)

Table 3.8: The distribution of causes of failure at sub-nominal voltage.

Batch number	Chips only failing at 1.1V	Failed power-up	Failed VMM calibration	Failed SROC calibration	Failed TTC calibration	Failed at least one test
1	93	12 (12.90%)	0 (0.00%)	9 (9.68%)	16 (17.20%)	56 (60.22%)
2	59	12 (20.34%)	7 (11.86%)	3 (5.08%)	2 (3.39%)	35 (59.32%)
3	154	15 (9.74%)	0 (0.00%)	8 (5.19%)	9 (5.84%)	122 (79.22%)
4	218	19 (8.72%)	4 (1.83%)	12 (5.50%)	19 (8.72%)	164 (75.23%)
Total	524	58 (11.07%)	11 (2.10%)	32 (6.11%)	46 (8.78%)	377 (71.95%)

Table 3.9: The distribution of causes of failure at sub-nominal voltage for the chips that pass at nominal voltage.

Each iteration of the TTC calibration procedure is a 20 ms test of the main functional features of the ROC processing logic but adapted to stress the TTC logic the most. Thus, the most failed chips were expected in this category. This is true for nominal voltage but not the case for the sub-nominal voltage testing where a clear majority of chips first fail during the ten tests. This can be explained by the relatively short test duration of the TTC calibration iteration (20 ms compared to the 5 or 10 s run time of a test) which is not enough to observe the meta-stability effects caused by the decreased fan-outs and the increased propagation, setup and hold times of the design cells. Also, the calibration functions do not apply a threshold on the minimum size of the obtained valid phase intervals. Even if only one phase produces valid results, the calibration is considered successful and that phase is used for the rest of the testing process. A chip with such channels will fail during the ten tests and thus will be included in that category. The sizes of the valid phase intervals decrease at sub-nominal supply voltages, as previously shown. This emphasizes the importance of the sub-nominal testing for the correct ROC sample characterization and therefore the good long-term ROC operation within NSW.

In Figures 3.19 and 3.20 the average, minimum and maximum power drawn from the two LDOs (one supplying the core logic - VDD and the other one the IO pads - VDDIO, respectively) by the mass-tested 1819 *good* ROC chips in different scenarios are plotted. These are measured using the power monitors of the fifth testing PCB version (see Figure 3.6). The solid lines correspond to the 1.2 V nominal supply voltage while the dotted ones to the 1.1 v sub-nominal voltage. The largest consumers are the ePLLs. The highest power draw is reached during test 9 (the one with the packet and trigger worst burst in a loop).

3.3 Conclusions of the ROC testing

The ROC ASIC required exhaustive digital and analog functional testing to validate its design and control the quality of its long-term results within the NSW TDAQ system. This chapter details the designed, implemented and deployed FPGA-based

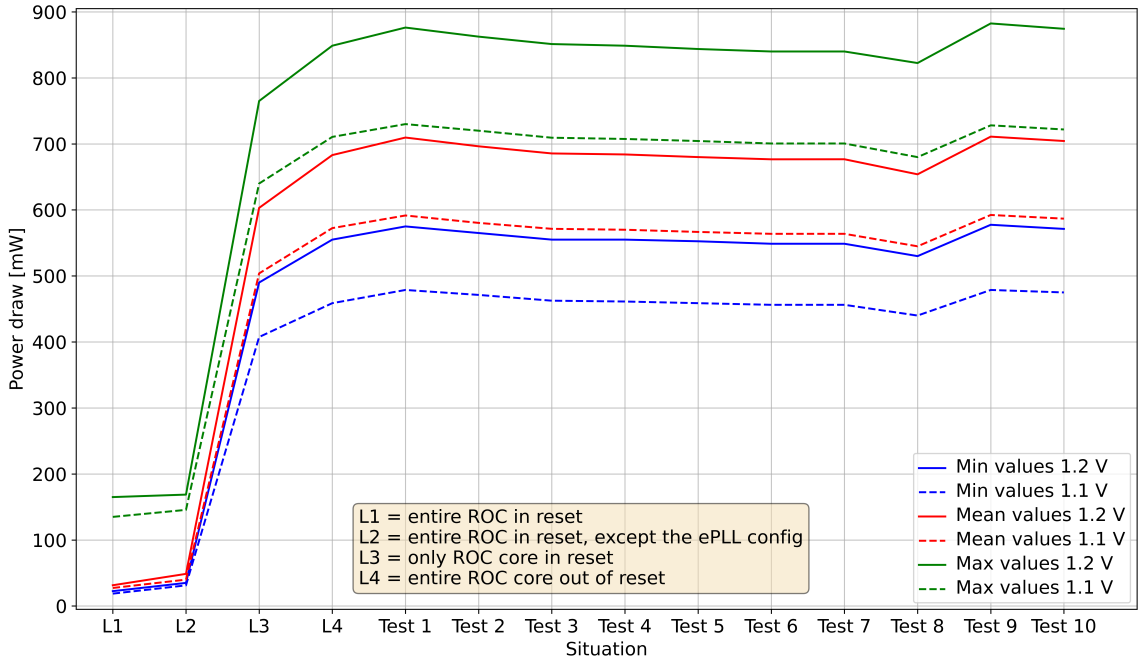


Figure 3.19: The average, minimum and maximum core logic power draw in different situations for the 1819 *good* chips.

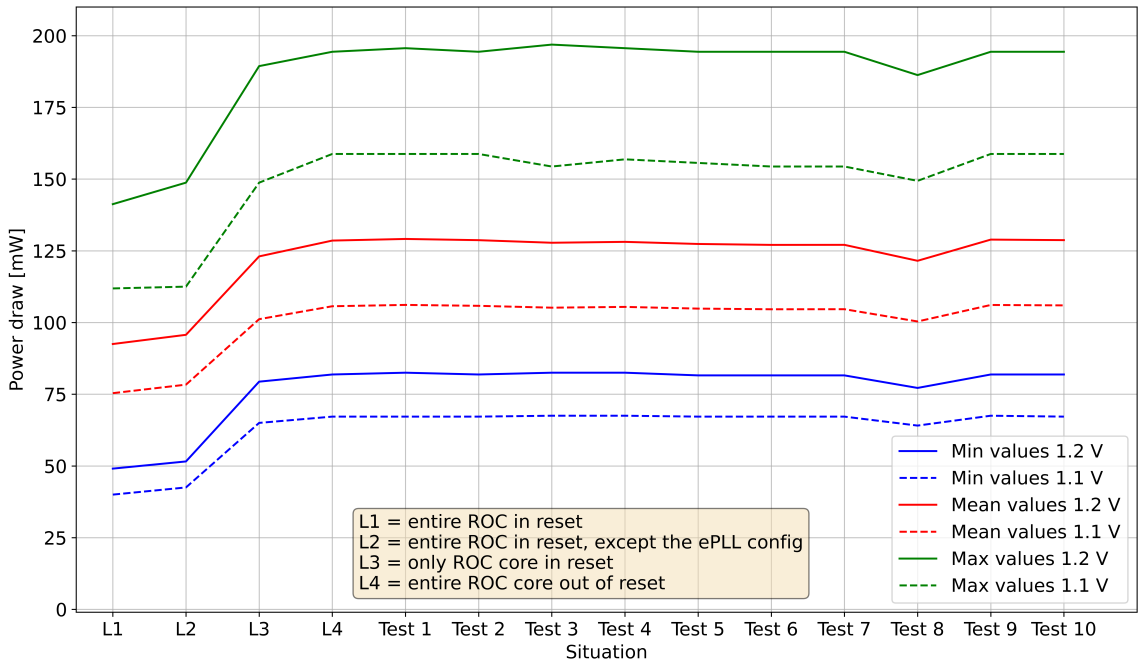


Figure 3.20: The average, minimum and maximum IO pads power draw in different situations for the 1819 *good* chips.

test setup for assessing the ROC packet processing digital logic. Both the ROC digital design and its mathematical model presented in Chapter 2 were successfully validated proving that the actual ASIC performance is limited only by the bandwidth of its output data links and the associated protocol.

In parallel, to also confirm the validity of the design, a team from University of Michigan developed a separate functional test setup based upon the same FPGA

evaluation board and the ROC accommodating PCBs. Their testing concluded that the analog part of the ROC required a redesign due to uncertain phases of the output clock signals upon power-up [61]. The digital design remained the same since it was validated. The redesign fixed the uncertainty issue [109].

The digital functional test setup presented in this chapter was prepared for mass-testing the ROC mainly by designing and employing the MicroBlaze-based subsystem and accelerating and improving the phase calibration for the communication lines between the FPGA and the ASIC. A method for synchronizing the clock and data signals per channel within the FPGA-based ASIC testing setup using delay lines specific to the Xilinx Ultrascale architecture was proposed. The deployed delay lines can also be used for channel jitter measurements as long as the sample time is sufficiently large. The proposed improved algorithm reduces the DUT input channels synchronization time by at least 70%.

Starting from December 2018, the ROC mass-testing occurred at Transilvania University of Brașov, Romania; Horia Hulubei National Institute for Research and Development in Physics and Nuclear Engineering (IFIN-HH), Măgurele, Romania and INCDTIM[2], Cluj-Napoca, Romania using the functional digital and analog (separate) test setups. The calibration of the high-speed data lines between the ROC and the FPGA was the main focus for improvements since the initial ROC testing yields were relatively small. Rather than redefining what a *good* ROC sample is, the effort was solely concentrated on improving the functional test setup. At Transilvania University of Brașov, the mass testing of 2677 chip samples (a total of approx. 10,000 chips were manufactured) obtained a satisfactory yield and proved its desired harshness [109]. To assure the quality of the *good* chips, a supply voltage 8.3% smaller than nominal was used. A large part of the failed chips experienced meta-stability. Since the design has assured correct functionality in these conditions from its implementation stage, these failures can be appointed to manufacturing defects but one can also take into account the worn-out socket of the testing PCB. The test setup is used to test more chips until the NSW required quantity is reached.

The ROC's radiation qualification test, presented in Chapter 4, used an adapted version of the presented digital functional test setup.

[2]Institutul Naţional de Cercetare Dezvoltare pentru Tehnologii Izotopice şi Moleculare

Chapter 4

Immunity to radiation-induced faults

The first section of this chapter presents the effects of nuclear radiation on electronic circuits. The basic techniques for mitigating them are included. The next section presents the implemented mechanisms for mitigating SEUs and Single Event Transients (SETs) within ROC. The ROC's functionality was tested in an environment with controlled neutron irradiation. The implemented and used test setup is thoroughly presented. Two ROC chips were subjected to neutron beams of 20, 22 and 24 MeV nominal energies, for over 39 hours, at an average flux of 9.5×10^5 n·cm^{-2}·s^{-1}, accumulating a dose equivalent to 8 months of operation within LHC at $L = 10^{34}$ cm^{-2}·s^{-1}. The digital part suffered 69 FF SEUs yet its operation was not affected by them, but data corruption within the memories occurred at a considerable rate. An in-depth analysis of the ROC's behavior in the irradiated environment is presented. New, theoretical scenarios are imagined, simulated and explained. The results are statistically analyzed and estimations are made for the operational environment. The chapter ends by presenting the conclusions related to these subjects. The research presented in this chapter was disseminated in [174].

4.1 Tolerance to radiation-induced faults

Nuclear radiation can disrupt the operation of electronic devices and systems and even permanently damage them through two phenomena: *ionization*[1] and *displacement damage*[2]. The radiation-induced effects are classified into two types: cumulative and Single Event Effects (SEEs). As their name suggests, the cumulative or total dose effects are proportional to the integrated flux of particles that were incident to the device. Eventually, they lead to the complete failure of the device. There are two cumulative radiation-induced effects: Total Ionizing Dose (TID) and displacement damage. SEEs are immediate results of ionization. They translate to current pulses and activation of parasitic structures within the semiconductor [55].

The displacement damage is expressed in the equivalent total number of 1 MeV neutrons incident per unit of surface area (i.e. 1 MeV eq. n $\cdot$ cm^{-2}). The induced

[1]The process of acquiring positive or negative charge by losing or gaining electrons, respectively, through collisions of sufficiently energetic particles with the ionized material [219].

[2]The process of inducing defects in the lattice of semiconductor atoms [55].

defects in the semiconductor crystal structure translate to altered device electrical characteristics (e.g. reduced conductivity and gain [219]).

The absorbed dose or TID is expressed in either rad or Gy (Gray), both being SI units. 1 Gy is the quantity of radiation that deposits 1 J of energy in 1 kg of exposed material. The rad is the older unit, still commonly used (1 rad = 0.01 Gy). As an ionizing particle passes through the semiconductor device it creates electron-hole pairs along its trajectory as it deposits energy. The electrons have greater freedom of movement [219] [55] while the holes accumulate in the oxides (e.g. the gate oxides). As the absorbed dose increases the electrical characteristics of the device change, e.g. for the n-MOS transistors the switch-on voltage threshold decreases while for the p-MOS transistors it increases [55].

The free electrons produced through ionization create a current pulse after the particle's passing. This pulse has different outcomes, some of them being *soft* (i.e. temporary affecting a signal) or *hard* (i.e. activating parasitic PNP and NPN parasitic structures in the semiconductor). Two typical types of *soft* failures are:

1. SET - induced rapid toggling (i.e. glitch) of the voltage level within a circuit net [52]. The effect on the system functionality depends on the net's function, e.g. the most disruptive SET happens on a clock tree branch.
2. SEU - induced change of state within a sequential element [55] caused by charge deposition. The value of the stored bit is flipped. The effect on the system functionality depends on the bit significance. The altered value persists until a new value is written or the device is powered off.

Three typical *hard* SEE failures are:

1. SEL - Single Event Latchup - formation of a low-impedance path between the power supply and the ground lines [55]. The electronic device becomes stuck in short-circuit until it is powered off. The high current heats the affected area and thus has the potential to permanently damage it [26].
2. SEGR - Single Event Gate Rupture - destructive and permanent formation of a conductive path between the gate and the channel of a power MOS FET (Field Effect Transistor) [43].
3. SEB - Single Event Burnout - activation of the parasitic bipolar transistor structure specific to a power MOS FET as a result of an induced high current [133]. The electronic device is permanently burned.

The on-site ICs used for space applications or physics experiments such as those in nuclear and particle physics are exposed to larger quantities of nuclear radiation compared to the ones for general purpose or off-site. In addition to the possibility of permanent damage, during operation, the *soft* SEEs can bring the circuits into states not considered in their design stage, called illegal states, that cause them to not function properly. SETs could bring the capturing sequential elements into meta-stability due to violated setup and/or hold timing requirements. Thus, their effect in digital circuits is similar to SEUs.

CERN defines two categories of electronic systems exposed to radiation: radiation-tolerant and radiation-hardened [55]. The radiation-tolerant devices are designed to operate within irradiated environments despite being vulnerable to radiation. They implement mitigation techniques for SEEs. The radiation-hardened devices are immune to all radiation-induced faults in the specified limits of their operating environment. They employ both technology and design mitigation measures. The ROC design is in between the radiation-tolerant and radiation-hardened categories as it

employs both design-based SEEs mitigation mechanisms and radiation-hardened technologies (i.e. the e-link pads and ePLL macroblocks) mixed with commercial off-the-shelf components (e.g. the SRAMs and the logic cells). As it will be shown in this chapter, its SRAMs are the design part most susceptible to radiation. In [120] it is shown that the used 130 nm CMOS technology does not require special shielding or hardness-by-design solutions for High Energy Physics (HEP).

The most common design SEEs mitigation techniques are: special transmission line encodings [21], Error Correcting Codes (ECCs) [21], FSMs with additional logic for recovering from illegal states, hardware and temporal redundancy, etc. All of these increase the design complexity, its die area[3] or resource usage[4] and consequently the power consumption and the cost.

TMR represents a classical hardware redundancy technique [183], being the fundamental form of N-modular redundancy, where N is three [181]. The N-modular redundancy, where N is a natural number greater or equal to three, usually odd, consists of replicating the system N times, feeding the same input to all systems and determine the correct output by majority voting. In the case of TMR, if one system fails, the other two can determine the correct output. If however two systems simultaneously fail, the voter will output the incorrect value. In the case of Penta Modular Redundancy (PMR), up to two systems can be erroneous simultaneously. In [181] TMR is compared with PMR: by implementing PMR instead of TMR reduces the unreliability by 7.5 times. The unreliability is $U = 1 - R$, where R is the reliability, defined as the probability of having no failure during a given operating time. The stated factor corresponds to an operating time $t = 0.1/\lambda$, where λ is the average failure rate.

4.2 Study of the ROC behavior in a neutron irradiation environment

The incident ionizing particles within the ROC's operating environment may cause SEEs in its logic. Thus, its specifications (see Chapter 2, Section 2.1) established the mitigation techniques to be applied. To evaluate their efficiency, a controlled irradiation test had to be performed. This section starts by presenting the broader implemented SEE mitigation techniques. Then, a version of the functional test setup adapted to stress and monitor the subject ROC ASICs while their silicon dies are hit by controlled incident neutron beams is detailed. The expected ROC behavior in these conditions is extensively described. Next, the real-world measurements and ROC's behavior while operating in the test setup under the incidence of the test neutron beams are presented, assessed and detailed. Based on the results, predictions are made regarding the chip's operation within the LHC environment.

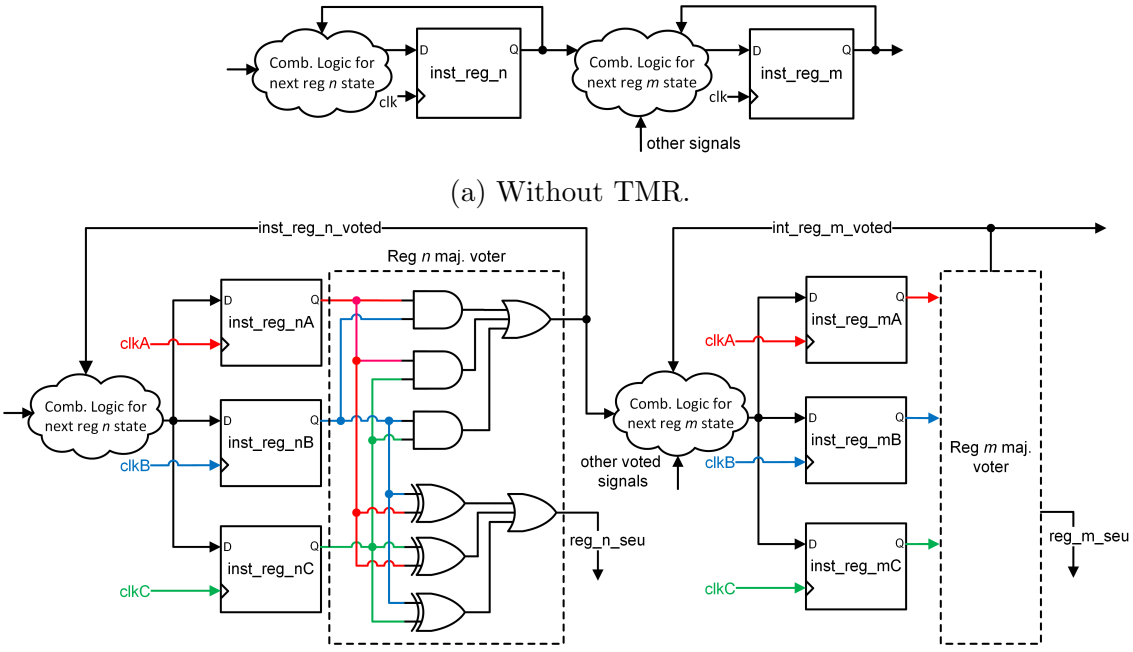
4.2.1 Implemented measures to mitigate radiation effects

TMR was implemented not only for the ROC's FSMs and configuration registers as the specifications mandate but for all its FFs as depicted in Figure 4.1. The rationale was that even if the configuration registers and the FFs within the FSMs

[3] in the case of an ASIC implementation

[4] in the case of an FPGA design

will always produce the correct output in response to the input stimuli due to TMR, corrupted input data might still drive them into unexpected states (i.e. in the used scenario) that cause unexpected behaviors and output. Examples of such unexpected behaviors being caused by radiation-induced bit alteration in the SRAM buffered data are presented in this chapter. In the digital part of the ROC the three instances of every FF are paced by distinct but (normally) identical (i.e. in frequency, fill factor, fan-out, rise and fall times and phase; even if the individual phases are adjustable) clock signals generated by the ePLL block (TMR BC clk and TMR RO clk in Figure 2.2 from Chapter 2, Section 2.3). Thus, SETs that could affect a slice of the ePLL or a clock tree, creating glitches on the corresponding clock signal, will not drive the digital part into an unknown, illegal state. In Figure 4.1 the three instances of the TMR clock signal are depicted with distinct colors. The digital logic that configures and monitors the ePLL blocks is also triplicated but a single clock signal paces every sequential element: the reference 40 MHz TTC BC clock signal. This logic cannot be paced by clock signals generated by the ePLLs that it controls. The SRAMs used as FIFO buffers in the ROC digital part do not use ECCs or TMR.



(a) Without TMR.

(b) With TMR.

Figure 4.1: Example of digital ROC logic without (a) and with (b) TMR.

As shown in Figure 4.1, for each triplicated FF the correct output is determined by a purely combinational majority voting module. The result is then supplied to the combinational logic (if any) that determines the next state for the next FF (also with three instances) in the pipeline and so on. If the FF has a feedback loop (e.g. a counter), the voted value is used instead of the value of the specific instance to correct a potential SEU. All the majority voting modules from the digital part also signal if the value of one of the instances does not match with the other two (e.g. the *reg_n_seu* and *reg_m_seu* signals from Figure 4.1). A large OR gate concentrates all these signals into one 1-bit wide signal that is used to increment the ROC's digital

part 8-bit SEU counter. The counter is capped at 255 and is cleared upon reading. The SEU OR signal is also fed to an output pad to be monitored by an external counter (highlighted as point 10 in Figure 2.17 from Chapter 2, Section 2.3.5). The digital logic controlling the ePLL blocks does not report its SEUs.

The triplication was performed using a not-finalized version of a Python script developed at CERN [137] and manual adjustments of the RTL code and SDC commands. The script receives as the input the Verilog file describing the module for which TMR must be implemented. The script generates a new Verilog file where each marked register and FF (i.e. each marked element declared as *reg* and described with an *always* block having one clock signal in the sensitivity list) is transformed into three instances with the same logic function as the original. The marking of a FF or register for TMR is achieved using a comment statement with special content during declaration (similar to how a net is marked for connection to the Xilinx ILA in an FPGA design). The resulting three instances have distinct names resulted by concatenating the original register or FF name with either *A*, *B* or *C*. For each one, a separate, adjusted (i.e. to the new name) *always* block is generated having the same logic as the original one from the input module. The script determines the width in bits for each case and passes the value to an instance of the parameterizable (in width) purely-combinational majority voting circuit. The data output of the majority voter keeps the original name from the input module (in Figure 4.1 the names of the resulting signals emphasize that they are voted). The SEU-related signals from all the majority voters within the module are ORed into one 1-bit wide SEU output signal. To indicate to the synthesis and PNR tools that the three instances must not be removed during logic optimization, for each triplicated register or FF, three SDC *don't touch* statements were manually included in the constraint file. The majority voters are not triplicated. Even if the next-state logic appears three times in the RTL code (i.e. in every *always* block instance) it is not protected from optimization by SDC commands. Furthermore, it uses signals described with single *assign* statements. Thus, the combinational logic between the majority voters and the next FFs is not considered triplicated.

This design can report false SEUs since the TMR is also applied to the capturing FFs that sample the serial data from the VMM3 ASICs, the ones that sample the TTC stream and the ones that sample the SCL and SDA I^2C lines for the digital ROC part. If at least one of these input data lines experiences transitions close to the edge (both edges for DDR) of the capturing clock signal, in the *forbidden* interval formed by the setup and hold requirements for that FF (e.g. due to an incorrect phase calibration or relatively high rise and fall times), it might drive it into meta-stability which will resolve to either logic 0 or 1 after a time with exponential distribution [118]. Since there are small skew differences even between the three capturing FF instances, some may resolve to logic '0' while others to logic '1'. When the three instances are not identical, an SEU pulse is produced on the output SEU pad and in parallel, the SEU counter is incremented. If this happens for at least one high-speed input serial line, in addition to decoding, misalignment and other errors the ROC SEU counter would indicate the maximum value (i.e. 255) upon each interrogation. The rate of this phenomenon is high enough that the counter saturates before the next read (relatively slow through I^2C). The higher the transition rate, the higher the probability of driving a capturing FF into meta-stability. That is why for the calibration of the signals within the FPGA-based

functional ROC test setup, the input data was varied by minimizing the occurrence of *comma* symbols (i.e. sending maximum size L0 and L1 packets back-to-back). The *K.28.5* comma symbols have relatively fewer transitions than other 8b10b words.

The I^2C SCL and SDA serial lines are relatively low-speed and have relatively long rise and fall times (relatively low slew rate), determined by the parasitic capacitance of the line, the used pull-up resistors and the driving/sinking currents of the devices that control the line. Thus, the corresponding ROC pads for these lines use hysteresis to filter glitches. Even so, meta-stability could still be induced in the capturing FFs. Since the I^2C slave expects low-speed signals, the sampled signals are filtered so that the meta-stability does not affect the functionality in the end but false SEU pulses might be registered. Thus, even if the capturing clock signals and the high-speed input serial data lines are calibrated, the SEU counter might have non-zero values after each I^2C transaction in conjunction with several pulses on the SEU pad. This behavior was observed during the digital design validation and mass-testing. The I^2C transition rate is much lower compared to the other data interfaces. Also, no transition happens while the line is idle in contrast to the VMM3 data lines which transmit back-to-back *commas* while idle. Thus, during each I^2C transaction, several pulses on the SEU output pad were expected (at maximum equal to the total number of rising and falling edges of the two I^2C signals). The internal SEU counter would indicate the same value. Since it is accessible solely through the I^2C interface, by reading it more false SEUs would have been produced. Therefore, during the ROC irradiation tests, the I^2C interface was not used for monitoring after the initial configuration so that no false SEUs were induced. The external FPGA-based SEU counter proved to be vital.

Another method for mitigation of the false SEU reporting behavior would have been a calibration procedure for the I^2C interface. This calibration was not implemented because of its complexity, its large impact on the per-sample run time during mass-testing and because the false SEUs do not affect the chip's functionality. During mass-testing, the recorded small non-zero values of the ROC internal SEU counter and the SEU output pulses during an I^2C transaction were attributed to the meta-stability of the capturing FFs and as a result they were ignored.

4.2.2 Test setup

The maximum annual flux[5] of neutrons with energies above 20 MeV (i.e. ultrafast neutrons) for the NSW was approximated to be 2×10^{11} n/cm^2/year [135], considering the LHC working at an instantaneous luminosity of $L = 10^{34}$ p/cm^2/s (i.e. during Run 2, see Chapter 1, Section 1.6). As a reference, this value was extracted from Figure 4.2 which depicts the annual TID and the flux of ultrafast neutrons for the NSW detector as a function of the distance from the LHC beam trajectory (i.e. radius r in Figure 1.4b, Chapter 1, Section 1.4). Both the TID and the ultrafast neutron flux are depicted on a base 10 logarithmic vertical axis and show a roughly linear decrease. The TID starts from approx. 500 Gy at 90 cm radius and reaches approx. 0.5 Gy at 430 cm while the ultrafast neutron flux begins at approx. 2×10^{11} n/cm^2/year and ends at 5×10^9 n/cm^2/year for the same distances.

[5]The flux represents the number of particles passing through a unit of surface area perpendicular to their trajectory in a unit of time [38], usually expressed in p $\cdot$ cm^{-2} $\cdot$ s^{-1}. The fluence (expressed in p $\cdot$ cm^{-2}) is the integration of the flux over time [38]. Thus, the flux is also called fluence rate.

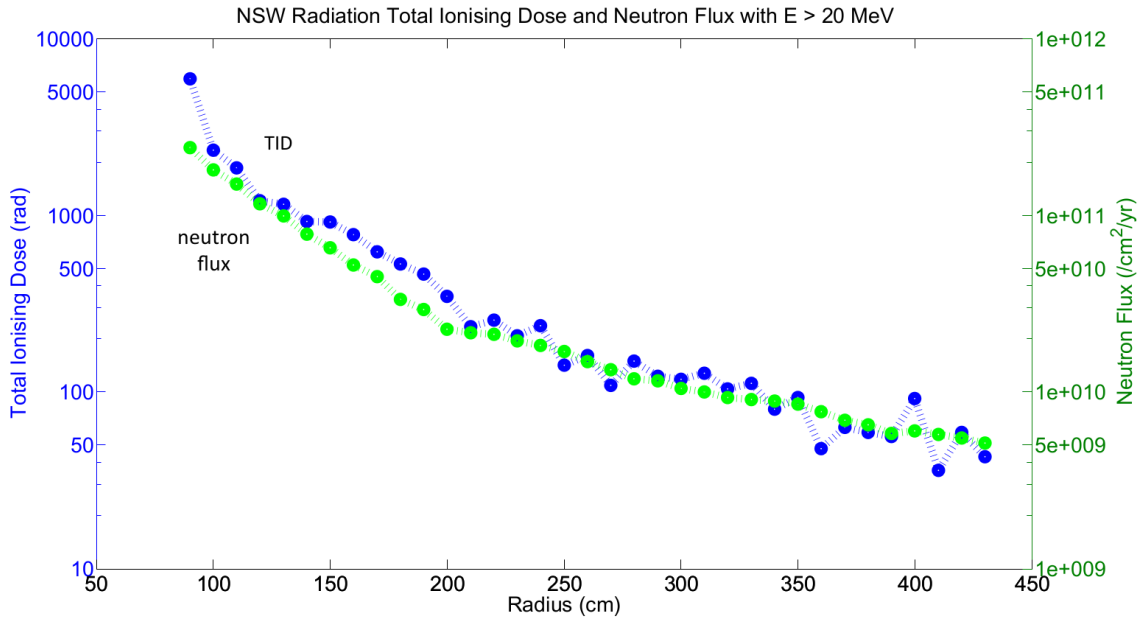


Figure 4.2: The NSW annual TID and ultrafast neutrons flux for $L = 10^{34}$ p/cm^{2}s as a function of the distance from the LHC beam trajectory [135].

As shown in Chapter 1, Section 1.6, after the current LHC upgrade process ends and the LHC is restarted, the luminosity will double. Then, in the following upgrade process, estimated for commissioning during 2026, the luminosity will be five to seven times larger. The ultrafast neutron flux will be at least multiplied with the same factors. The term *at least* is used since other nuclear reactions might contribute to the flux.

The Tandem accelerator from the National Centre for Scientific Research (NCSR) Demokritos, Athens, Greece, produces beams of neutrons with energies between 0.1 and 25.7 MeV [41] using various nuclear reactions. A total neutron fluence comparable to the maximum NSW annual one can be achieved in several tens of hours. Even if the energy spectrum of the produced beams does not completely match the spectrum within the NSW and the parasitic nuclear interactions are not very well studied or documented, the Demokritos facility was considered to be appropriate by the NSW Group for the radiation qualification process of the ROC and the other context ASICs. This subsection describes the used test setup, its configuration and the expected ROC behavior under irradiation.

The functional digital test setup used for design validation and chip mass-testing, presented in the previous chapter, was adapted for the irradiation tests. It was considered ideal since it emulates the ROC's NSW context and covers all possible states and functional features for the packet processing logic being highly configurable. In addition, it can detect any incorrect bit of any output packet while the ROC functions in real-time at its targeted frequencies.

The ultrafast neutron beams from the Tandem facility are produced through the ${}^{3}_{1}\mathrm{T} + {}^{2}_{1}\mathrm{D} \longrightarrow {}^{4}_{2}\mathrm{He} + {}^{1}_{0}\mathrm{n}$ nuclear strong interaction, where the D and T are Deuterium and Tritium, both isotopes[6] of hydrogen (${}^{1}_{1}\mathrm{H}$), He is helium and n is the resulted neutron. In Figure 4.3, the applied method of testing chips in the resulted beam

[6]variant of the same chemical element (i.e. the same number of protons in the nucleus) occupying the same cell in the periodic table of elements but having a different number of neutrons.

is depicted. The deuteron (i.e. D, the nucleus of Deuterium ($^2_1$D)) beam hits a tritiated (T) Titanium (Ti) target, the nuclear interaction happens and the beam of ultrafast neutrons (n) results. Multiple PCBs with the Devices Under Test (DUTs) are placed in the beam, in a plane normal to it. The PCBs are squeezed together as much as possible without making contact to maximize the flux of neutrons. The *Liquid Counter* is a scintillator[7] used for monitoring the stability of the accelerator during its runs. The order of the DUTs can be changed between these runs until the desired fluence is reached for each one.

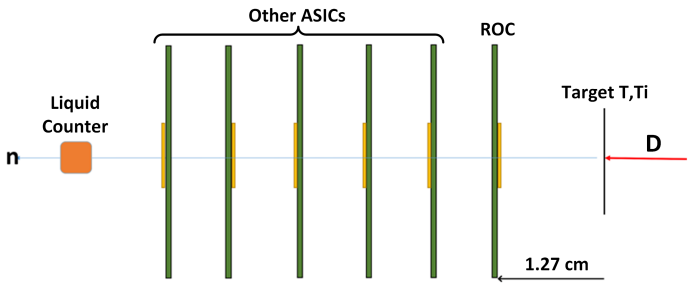


Figure 4.3: The test setup used for irradiating multiple DUTs, the ROC testing board placement for the first four irradiation test runs and the nuclear strong interaction that produces the ultrafast neutrons at the TANDEM accelerator.

In Figure 4.4 an overview of the test setup used for the ROC radiation qualification is presented, showing the placement of the ROC testing PCB relative to the beam, its interface with the shielded FPGA evaluation board and the remote access host computer used for control, communication and data logging. The interaction area is detailed in Figure 4.5 when only the ROC's testing PCB was mounted. The mass-production testing PCB with the open-top socket (depicted in Figure 3.6) hosting the BGA packaged ASIC (as illustrated in Figure 2.18d) was positioned so that the silicon die was directly exposed to the neutron beam as close as possible to the source (i.e. 1.27 cm for the first four beam runs). The testing board was interfaced with the FPGA evaluation board through a 1.8 m long VITA 57.1 FMC HPC cable. The second LPC FMC connection was not used, as shown in Figure 4.5 since it exclusively carries signals from the ROC's analog part which was not the subject of the tests. The FPGA evaluation board was positioned outside the beam trajectory, being shielded by lead bricks filled with paraffin and was interfaced through JTAG to USB and UART to USB with a host computer with remote internet access. A terminal application logged the UART data into text files and acted as the main user interface to the functional test setup. The Xilinx Vivado Lab program was used to upload the firmware bitstream file to the FPGA and to download the ILA captured signals samples via USB-JTAG.

The FPGA design Xilinx ILA's buffer depth (2^{14} samples) was split into 2^6 or 2^7 windows of 2^8 or 2^7 sample depth, respectively. The Xilinx ILA fills a window with consecutive samples of pre-selected signals from the FPGA design for each trigger pulse. Once all the buffer space is filled, its content is flushed to the host computer through the USB-JTAG interface. During this transfer, the new trigger pulses are discarded and thus events are lost because there is no guarantee that there is enough buffering space for a new full sample window. More details are presented in Chapter

[7]material that produces scintillations (i.e. flashes of light) when is struck by ionizing particles.

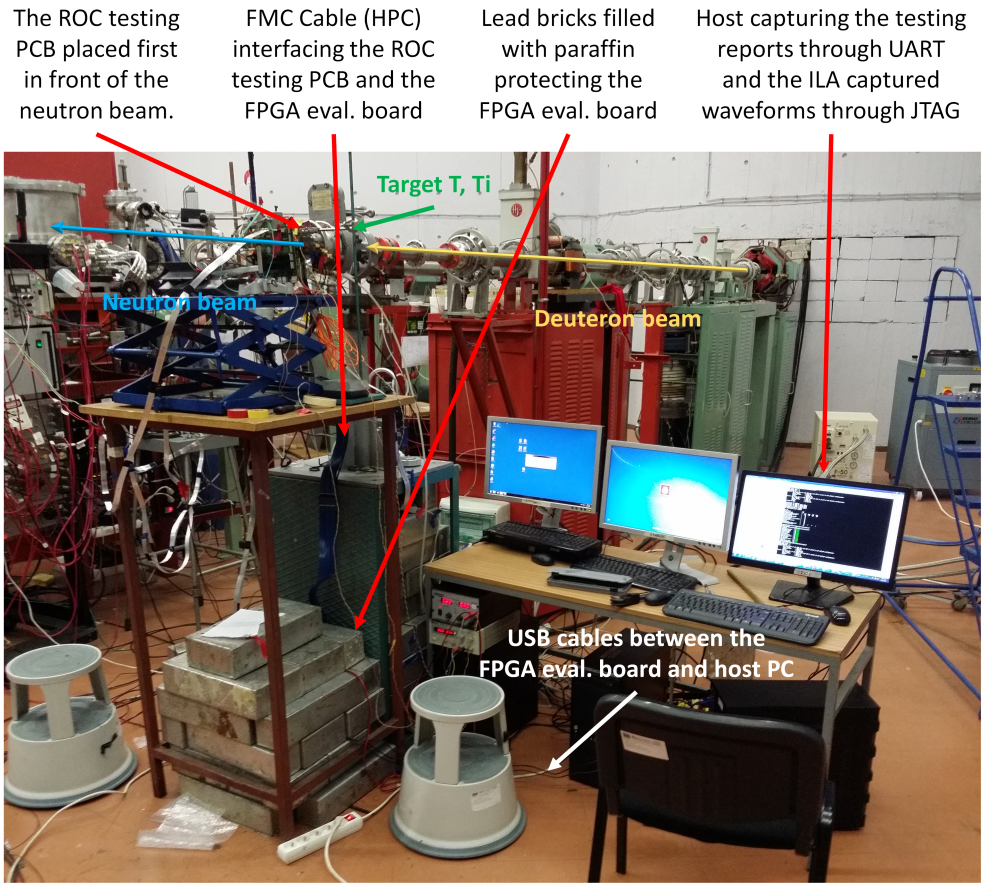


Figure 4.4: The test setup used for testing the ROC in the ultrafast neutron beam produced by the Tandem NCSR facility.

5. For the ROC's irradiation testing, the ILA trigger condition was set to be an OR of all the error flags produced by the four SROC Capture output data analyzers (see Chapter 3, Section 3.2.2). An offset was set so that the moment when the trigger happened is placed in the middle of the ILA window. Thus, the cause of the produced error and its short-term effect could be examined. When the transfer is complete the ILA trigger must be reactivated to be able to capture new windows. Thus, a TCL script was developed and executed in a loop within Xilinx Vivado Lab that saved the ILA waveform data transmitted by the FPGA board and re-enabled the ILA triggering once the transfer was complete.

Rather than looping through the 10 tests used for mass testing (see Chapter 3, Section 3.2.2), the test setup continuously injected and analyzed data into and from the ROC. The pseudo-code of the MicroBlaze C program is depicted in Algorithm 3 and the called functions, other than those already presented in Table 3.2 from Chapter 3 Section 3.2.2, are detailed in Table 4.1. The program initializes the MicroBlaze peripherals and the testing PCB, calibrates the input and output FPGA serial lines, similar to the method described in Chapter 3, Section 3.2.3, then initializes the ROC and starts the VMM3 and TTC stream emulators while the ROC maintains its default (i.e. from reset) configuration. Every 10 seconds (i.e. iteration) a report of the encountered errors in the ROC output data from all four SROCs is provided through UART. Counters were implemented to record the amount of raised error flags by type. When a flag is raised, the corresponding counter is incremented and

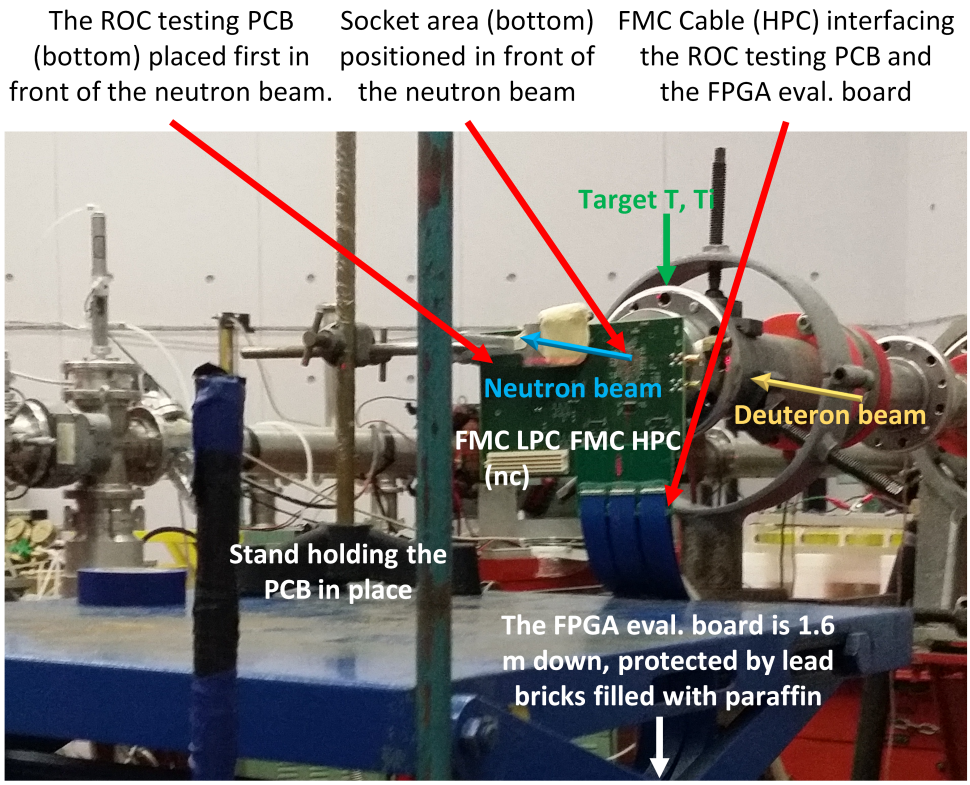


Figure 4.5: Detailed view of the interaction area within the test setup depicted in Figure 4.4.

the flag cleared.

Function	Description
read_TB_status()	Updates a global test-bench status data structure with the flags and measurements related to the ROC behavior. Increments the corresponding error counters.
print_TB_status ()	Prints the flags, counters and measurements related to the ROC behavior.
print_power()	Prints the instantaneous ROC power consumption.
check_TB_status()	Checks the status global data structure for errors.
clear_TB_errors()	Issues a signal which clears all the reported error flags (the error counter values are untouched).

Table 4.1: Descriptions of the specific functions called in the irradiation testing Algorithm 3.

The VMM3 emulators were configured to generate packets with constant hit content but unique to each one to easily identify the origin even when bits are altered. Consequently, there is no need for resetting the chip and the test environment after the first encountered hit content-related error. After the chip initialization, no other transfers happen on the I^2C interface of the digital part. The status registers of the ROC were not monitored because further transfers on the I^2C interface of the digital part could generate false SEUs, as described in Section 4.2.1. The real SEU

Algorithm 3 The irradiation testing program pseudo code. The specific functions are detailed in Table 4.1.

```
1: if init_&_test_MB_periph() then return 1
2: chip_id ← read_chip_id()                                          ▷ global variable
3: result_calib_out_0 ← calib_fpga_out_ser_data(shift = 0)
4: result_calib_out_1 ← calib_fpga_out_ser_data(shift = 1)
5: result_calib_out ← max(result_calib_out_0, result_calib_out_1)    ▷ global variable
6: result_calib_in ← calib_fpga_in_ser_data()   ▷ global variable; cascaded delay lines
7: ttc_stream_start ← determine_ttc_stream_start()                   ▷ global variable
8: result_calib_ttc_0 ← calib_ttc_stream(shift = 0)
9: result_calib_ttc_1 ← calib_ttc_stream(shift = 1)
10: result_calib_ttc ← max(result_calib_ttc_0, result_calib_ttc_1)   ▷ global variable
11: if not result_calib_out or not result_calib_in or not result_calib_ttc then
12:     result ← manual_mode()
13:     if result then return 1
14: result_pu ← power_up_ROC()
15: result_tb_config ← write_tb_config()
16: if result_pu or result_tb_config then return 1
17: timer_finished ← 0   ▷ global volatile variable incremented in the timer interrupt handler
18: start_test()   ▷ starts the test with the ROC in its default configuration; starts the timer
19: iteration ← 0
20: while true do
21:     read_TB_status()
22:     if timer_finished then
23:         iteration ← iteration + 1
24:         print iteration
25:         print_TB_status()
26:         print_power()
27:         timer_finished ← 0
28:     result ← check_TB_status()
29:     if result then clear_TB_errors()
30: return 0
```

occurrences are recorded by sampling the ROC SEU output pad only.

Bit flips in the ROC's SRAMs could corrupt the output data, cause the loss of data and drive the FSMs into unexpected (for the used scenario) but valid (i.e. designed) states. Since the ROC's buffers operate as FIFOs, it does not matter if a bit is flipped at an address that does not contain data yet or that contains data already processed. When the FIFO's write pointer reaches the affected address, the logic simply writes the new data. The read pointer will not access that address before the new data is written. The filled region of the memory is considered the group of addresses containing data not yet processed, i.e. the addresses between the write and read pointers. This region does not cover the same memory area all the time. Instead, it is looping over all the address space as data is being written and read from the FIFO. The size of the filled region (i.e. the occupancy level) is usually more significant in the VMM Capture FIFOs than in the SROC and BC FIFOs, being proportional with the L1 trigger latency, the input packets rate and their size.

Most bit flips that could affect the ROC's behavior and data will happen in these FIFOs. During the irradiation tests, the occupancy levels of all ROC FIFOs were raised as much as possible without inducing data loss to increase the chance that the ROC's behavior and output data will be affected.

The highest L1 trigger latency supported by the test environment at that time (i.e. 180 μs) was selected. The input packets were injected at an average 1.37 MHz rate. The non-empty input packets contained an average of $\bar{n} = 8.174$ hit words. The average probability (i.e. across all eight emulators) of null-event VMM3 packets was $p = 0.5137$ resulting in an average uncoded effective throughput of 196.1 Mbps per VMM3 emulator. The effective uncoded throughput represents how much of the available 512 Mbps uncoded (640 Mbps 8b10b encoded) bandwidth is used for sending packet data. The ROC was in its default configuration: each SROC dealt with L0 data from two VMM Capture channels and the resulted L1 data is transmitted at the maximum speed of 512 Mbps uncoded. The resulted output events are larger than the corresponding input ones because they aggregate data from the two connected VMM3 emulators. Also, supplementary trailer words are generated for all of them, except for the L1 null-event packets. In these conditions, no L0 packets were truncated or discarded and the average uncoded effective throughput for the ROC output data streams was 418.36 Mbps (81.71%), not considering the SOP and EOP symbols. In Figure 4.6, the typical evolution of the VMM Capture and SROC FIFOs occupancy levels obtained in these conditions is depicted. The emphasis is on the steady-state averages obtained once the issue of L1 triggers begins. The maximum, average and minimum cases are depicted for the VMM Capture channels. The average occupancy level for the VMM Capture FIFOs was 1227.43 addresses out of the available 2049 (59.90%). The average occupancy level for the SROC and TTC FIFOs was 21.23 (1.04%) and 0.21 (0.17%) addresses, respectively. The average ROC FIFO occupancies by word types and in total are detailed in Table 4.2. For each type of word, the corresponding number of occupied SRAM bits is also listed. These values were used for computing the cross-sections of the SRAM-related errors in Sub-section 4.2.3.

Average occupancy	**Words**	**%**	**Bits**	**%**
Header words in one VMM Capture FIFO	246.6	12.04	4192.2	6.2
Hit data words in one VMM Capture FIFO	980.83	47.87	32367.39	47.87
Total for one VMM Capture FIFO	**1227.43**	**59.90**	**36559.59**	**54.07**
Null-events in one SROC Packet FIFO	0.4069	0.02	6.9175	0.01
Header words in one SROC Packet FIFO	1.1351	0.06	37.4577	0.06
Trailer words in one SROC Packet FIFO	1.1351	0.06	37.4577	0.06
Hit words in one SROC Packet FIFO	18.5563	0.91	612.3583	0.91
Total for one SROC Packet FIFO	**21.2334**	**1.04**	**694.1912**	**1.03**
L1 triggers in one TTC FIFO	0.2158	0.17	6.474	0.17
Total for all ROC FIFOs	**9905.235**	**39.46**	**295279.38**	**35.71**

Table 4.2: The average ROC FIFOs occupancies for the used test configuration.

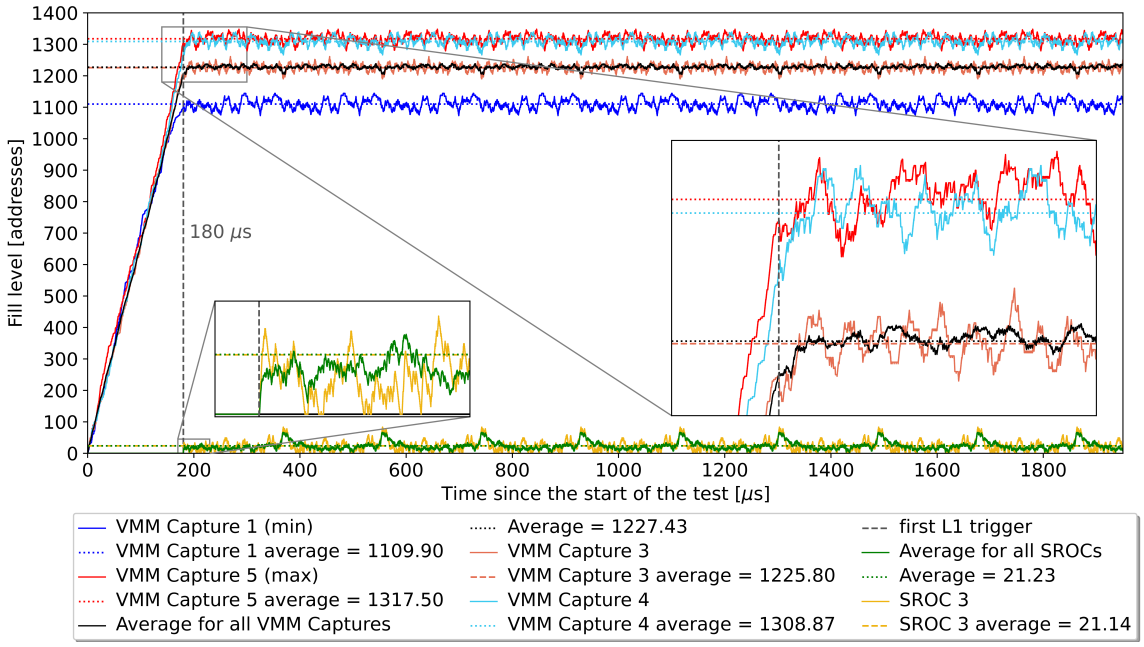


Figure 4.6: Typical evolution of the fill levels for selected VMM Capture and SROC FIFOs while the system runs with the same configuration as in the irradiation test.

To achieve higher occupancies in the ROC's FIFOs and, consequently, more bit flips that could affect the ROC operation, the output links should be saturated or limited to lower bandwidth, causing the SROC FIFOs to fill up. The SROC Packet Builder FSMs will process data only when free memory is available in the SROC Packet FIFO, causing the received L1 triggers to fill the TTC FIFOs and the input data to fill the VMM Capture FIFOs. Trigger and data loss will occur, causing the rise of error flags within the SROC Capture output data analyzers (see Chapter 3, Section 3.2.2). In the irradiation test setup, the aim was the recording only of the potential errors induced by radiation, hence the used configuration.

Since TMR was implemented for all the ROC's FFs, no misalignment, decoding, encoding and output protocol syntax errors were expected. Also, no change in the configuration or FSMs reaching illegal states (i.e. unexpected from the design phase) should happen. The ROC logic should not be affected when SEUs happen in its FFs. However, loss of data and errors in the output packets, both caused by SEUs in the SRAMs were expected. The test environment is designed to detect any error in any ROC output packet. Within it, the errors are classified by the affected packet field or encountered behavior. In each of the following marked paragraphs, one type of error that can be caused by altered SRAM bits is detailed, its source is determined alongside the caused data loss and potential for recovery assuming no other unrelated error affects the altered packet. These types of errors cover all the packet bits but are not mutually exclusive. An altered packet bit can cause the rise of multiple types of errors. Also, the mentioned error causes do not take into account the alterations of the auxiliary bit, which signals the end of the packet (i.e. the bit with index 32 in Figures 2.14 and 2.16). Thus, the word type can be confused, causing multiple other errors. Given the distribution of occupied bits from Table 4.2 and considering that the probability for the SRAM latches to suffer SEUs is equal between them, the order of the following paragraphs depicts the sorted error list from the most to the least expected.

The hit content error indicates an unexpected value in any of the fields of a hit word related to detector data (i.e. it covers 28 bits out of the total 32 bits of a hit word). The 3-bit VMM Capture number from the output hit words is covered by the VMM error type. The flags of the input hit words are not covered. It is caused by bit flips in the VMM Capture or SROC Packet FIFOs. If the packet checksum is correct then the affected hit words were altered in VMM Capture FIFO, else the flip happened in SROC Packet FIFO. Assuming equal probabilities for each of the covered bits to experience flips and that only one of the bits is altered at a time, the chance that the affected hit word will indicate wrong parity is $22/28 = 78.57\%$. This error is unrecoverable and the data associated with the corresponding detector channel for the selected BC is lost.

The parity error indicates a hit word with incorrect parity. It covers the same fields as the hit content error, except the 6-bit VMM3 channel number. It also takes into account the BCID value from the packet header. This type of error is usually caused by hit content or OrbitID & BCID errors. If the packet checksum is correct, then the error is caused by a bit flip in the VMM Capture FIFO, otherwise in the SROC Packet FIFO. It can also be caused by an altered parity bit in an otherwise untouched hit word. If the root cause is an altered BCID header value, then all the hit words of the affected packet will indicate wrong parity. If the BCID can be corrected, the parity will be right and the data will not be lost.

The *VMM missing* error is triggered when a trailer word indicates missing data when no data should be missing. There are many reasons for missing data, as presented in Chapter 2, Section 2.3.2. For the used test environment configuration, the following scenarios could cause this type of error:

1. an altered VMM3 overflow flag in the L0 header (i.e. a set V bit);
2. an altered truncate flag (i.e. set T bit) in at least one of the input hits;
3. at least one of the VMM Capture FIFO overflowed due to previous data errors;
4. an altered L0 header flushed by the SROC due to incorrect parity;
5. an altered *VMM missing* trailer field (also causing a checksum error) and
6. an altered L1 trigger causing the SROC to search for an incorrect BC.

Bit flips within the VMM Capture FIFOs cause scenarios 1, 2 and 4 while bit flips within the SROC Packet and TTC FIFOs cause scenarios 5 and 6, respectively. In the latter scenario, if the searched BC becomes older than the original, it could also cause the accumulation of L0 data in the VMM Capture FIFOs. This behavior could lead to them overflowing (causing more data to be lost and more *VMM missing* errors to be recorded - see scenario 3). The SROC could lose the synchronization between the L1 triggers and the L0 data. In this situation, the chip will require a soft or hard reset. The data loss in all these scenarios cannot be recovered.

The checksum error indicates an incorrect checksum value in a trailer word. The checksum is computed over the entire packet data. Thus the error is usually a result of the other types of errors but can also be caused by an altered bit within the checksum field itself. This error can be produced only by bit flips in the SROC Packet FIFO. If the underlying error that also caused the checksum to become incorrect can be solved, then the checksum will become valid. If not, the packet must be discarded.

The OrbitID & BCID error indicates unexpected BC information in the packet header. Since the SROC discards all L0 packets with incorrect BC information from the VMM Capture FIFOs, this error can be caused by bit flips in the L1 headers buffered in the SROC Packet FIFO or by altered L1 triggers from the TTC FIFO. If the error appeared in the SROC Packet FIFO, a checksum error is also produced. If the BCID value is affected, all the associated hit words will rise parity errors. The back-end NSW TDAQ system could correct the BC information and then recompute the checksum and the parity of each hit word if it maintains a history of the selected BCs of interest. It could also discard the packet with the unexpected BC information but a correct checksum field produced by an altered L1 trigger from the TTC FIFO.

The VMM error indicates a hit word with an incorrect 3-bit VMM3 identifier. The hit word appears to be produced by a VMM3 that is not associated with the SROC or by an associated VMM3 different than in reality. Since this field is present only in the output hit words, this error can be produced only by bit flips in the SROC Packet FIFO. The parity-bit of the hit word does not cover this field. A checksum error is also produced. In some cases, the affected packet can be recovered considering that the SROC outputs the hit words from the associated VMM Capture FIFOs starting from the one with the smallest identifier in ascending order. Knowing the associated VMM Capture channels and the rest of the packet content, the back-end NSW TDAQ system could infer the correct VMM3 identifier.

The L1ID error signals a header or null-event with an unexpected L1 trigger count value. This error can be caused by bit alterations in the TTC or SROC Packet FIFOs. For the packets starting with headers containing a damaged L1ID value, the bit alteration took place in the TTC FIFO if the checksum is correct and in the SROC Packet FIFO otherwise. For the null-events that indicate an incorrect L1ID value, it is impossible to determine exactly where the bit alteration occurred, since they do not contain a checksum field. The back-end NSW TDAQ system could determine the correct L1ID value by looking at the BC information and/or the L1ID values of the previous and following packets.

The No. hits error indicates an incorrect number of hit words being reported in a trailer. This error can be caused only by bit flips in SROC Packet FIFO. A checksum error is also produced. The affected packet can be salvaged by counting the actual number of contained hit words, replacing the length value in the trailer word and recomputing the checksum.

The ROC ID error signals a null-event with an incorrect value for the ROC ID field. This error can be caused only by bit flips in the SROC Packet FIFO. Since the null-event packets do not contain parity or checksum fields, the error is not detectable by simple packet inspection. The role of the null-event packets is to signal L1 triggers that had no matching hit data from the associated VMM3 chips. It is essential to correctly identify the source ROC to infer the VMM3 chips and therefore the detectors that did not produce any data for the BC. If the back-end NSW TDAQ system knows the source of the packet, the data can be recovered by substituting the actual ROC ID value.

4.2.3 Irradiation tests results

In this section, the parameters, measurements and computed characteristics for the ROC tests within the ultrafast neutron beam produced at NCSR are presented alongside rationales and ROC behavior descriptions. SEUs were recorded both in the ROC digital part FFs and SRAM latches. The section ends with simulated scenarios in which bit flips in the SRAMs cause data loss and de-synchronization between the L1 triggers and the L0 data.

Two validated ROC sample chips were subjected to nine ultrafast neutron beam runs which are detailed in Table 4.3. The first chip sample was used for the first four runs and the second chip for the last five. The first ROC sample was always the closest ASIC from the source, while the second sample was placed as the fourth (run 5), the third (runs 6, 7 and 8) and the second (run 9) ASIC from the source. The nominal energy of the neutron beam was varied between 20, 22 and 24 MeV. The neutron flux was estimated for each run by the accelerator operators, as a function of the distance and the interposed materials between the source and the ROC die. The stated uncertainty is ±10%. Details on the estimation of the flux uncertainty of a similar setup are given in [116]. Based upon the run-time of each run, the neutron fluence was computed as the product of the flux and the run-time. Then the resulted fluence was compared with the previous maximum LHC annual fluence of 2×10^{11} n·cm^{-2} [135] resulting in the stated equivalent LHC run-times. The number of neutrons that hit the ROC die is estimated by multiplying its surface area with the fluence.

The FF SEUs counts in each beam run are listed in Table 4.4. In total 69 FF SEUs were recorded from the 18458 FFs of the ROC digital part. Almost half of the detected FF SEUs happened in run 3 when the nominal energy of the neutrons reached 24 MeV and the ROC die was closest to the neutron radiation source. By contrast, in run 1, the detected FF SEUs represent only 16% of the total number of recorded FF SEUs in the experiment. The same chip sample is used in the same position, being hit by a beam with a smaller flux, for three times the run-time totaling more than twice the fluence but with neutrons reaching only 20 MeV nominal energy. Similarly, in run 2, for a comparable fluence of neutrons with 20 MeV nominal energy, five times fewer FF SEUs were recorded. The ultrafast neutron energy spectra at the ROC level for runs 1 (20 MeV) and 3 (24 MeV) are illustrated in Figure 4.7. They resulted from Neutron Source Description (NeuSDesc) [60] simulations tuned by the accelerator operators and validated with measurements [134] [194]. In the case of the 20 MeV nominal energy, the actual average energy is around $19.6^{+0.4}_{-0.34}$ MeV while in the case of the 24 MeV nominal energy, the average actual energy is around $23.7^{+0.4}_{-0.45}$ MeV. These are the only two cases for which an energy spectrum was provided. Therefore, in this thesis when energy is quoted, it refers to the nominal value which differs slightly from the average one. For each run, the ratio between the recorded FF SEUs and the neutron fluence is divided by the number of FFs within the ROC digital part, resulting in the FF SEU cross-sections σ_{FF} from Table 4.4.

The confidence intervals are computed after the methodology presented in [38], considering a fluence uncertainty of ±10% and a 95% confidence level. The working hypothesis is that the SEUs are random following a Poisson distribution (i.e. the arrival of an event is independent of the event before). The methodology depicts the calculation of the upper and lower bounds for the number of events, given the

Run	Nom. E [MeV]	Dist. from source [cm]	Pos.	Flux [n·cm^{-2}·s^{-1}]	Time [s]	Fluence [n·cm^{-2}]	LHC [months]	Neutrons incident on the ROC die
1	20	1.27	1st	(1.5 ± 0.15)E+6	38,160	(5.7 ± 0.57)E+10	3.42 ± 0.34	(1.28 ± 0.13)E+10
2	20	1.27	1st	(1, 2 ± 0.12)E+6	19,560	(2.4 ± 0.24)E+10	1.46 ± 0.15	(5.48 ± 0.55)E+9
3	24	1.27	1st	(1, 8 ± 0.18)E+6	12,240	(2.2 ± 0.22)E+10	1.35 ± 0.13	(5.07) ± 0.51)E+9
4	22	1.27	1st	(2, 2 ± 0.22)E+6	7,200	(1.6 ± 0.16)E+10	0.94 ± 0.09	(3.52 ± 0.35)E+9
5	20	4.56	4th	(1.3 ± 0.13)E+5	36,120	(4.7 ± 0.47)E+9	0.28 ± 0.03	(1.05 ± 0.11)E+9
6	20	6.9	3rd	(5.6 ± 0.56)E+4	13,800	(7.7 ± 0.77)E+8	0.05 ± 0.005	(1.73 ± 0.17)E+8
7	20	4.4	3rd	(1.4 ± 0.14)E+5	1,860	(2.5 ± 0.25)E+8	0.02 ± 0.002	(5.67 ± 0.57)E+7
8	24	4.4	3rd	(2.7 ± 0.27)E+5	2,760	(7.5 ± 0.75)E+8	0.05 ± 0.005	(1.69 ± 0.17)E+8
9	24	2.4	2nd	(8.4 ± 0.84)E+5	10,260	(8.6 ± 0.86)E+9	0.52 ± 0.05	(1.94 ± 0.19)E+9
				Total	141,960	(1.35 ± 0.13)E+11	8.08 ± 0.81	(3.03 ± 0.3)E+10

Table 4.3: The parameters of the nine ROC irradiation test runs referencing the maximum ultrafast neutron NSW fluence previously obtained at an LHC luminosity of L = 10^{34} p·cm^{-2}·s^{-1}.

recorded number of events and the confidence level. The smaller the number of recorded events is, the larger the resulted interval. Based on the fluence uncertainty and the upper and lower deltas for the number of events, the upper and lower FF cross-section values were computed. In Table 4.5, the obtained FF cross-section averages over the runs with the same beam energy, including their confidence intervals, are listed. As a reference, in [41], the first version of the VMM ASIC, manufactured in the same technology, was tested in a similar setup at NCSR and resulted in a cross-section of $4.1^{+0.7}_{-0.7} \times 10^{-14}$ cm^2· bit^{-1} in a 20 MeV neutron beam.

The determined cross-sections should be considered as effective for the used setup. They are not due strictly to the quasi-mono energetic neutrons of energy in the range 20 – 24 MeV provided by the accelerator. The resulting significant parasitic neutron spectrum extending from high down to very low energies should be taken into account. These neutrons originate from Deuteron-induced reactions within the structural materials of the target and Deuteron break-up in Tritium (above 20 MeV). In [194] full range neutron energy spectra are shown for this setup. The neutrons result through the same reaction as during the ROC irradiation tests but have only 15.3 MeV nominal energy. Between 14 and 16 MeV, the patterns are similar to the ones from Figure 4.7. The graphs show significant values between 0.5

Run	FF SEUs	FF SEUs lower	FF SEUs upper	σ_{FF} [$\times 10^{-14}$ cm^2·bit^{-1}]
1	11	5.49	19.68	$1.04^{+0.85}_{-0.56}$
2	6	2.20	13.06	$1.33^{+1.59}_{-0.88}$
3	31	21.06	44.00	$7.45^{+3.45}_{-2.8}$
4	10	4.80	18.39	$3.46^{+2.98}_{-1.93}$
5	3	0.62	8.77	$3.47^{+6.71}_{-2.84}$
6	1	0.03	5.57	$7.05^{+32.3}_{-7}$
7	1	0.03	5.57	$21.5^{+98.4}_{-21.4}$
8	1	0.03	5.57	$7.21^{+33}_{-7.17}$
9	5	1.62	11.67	$3.14^{+4.23}_{-2.21}$
Total	**69**	**53.69**	**87.32**	$\mathbf{2.78^{+0.92}_{-0.82}}$

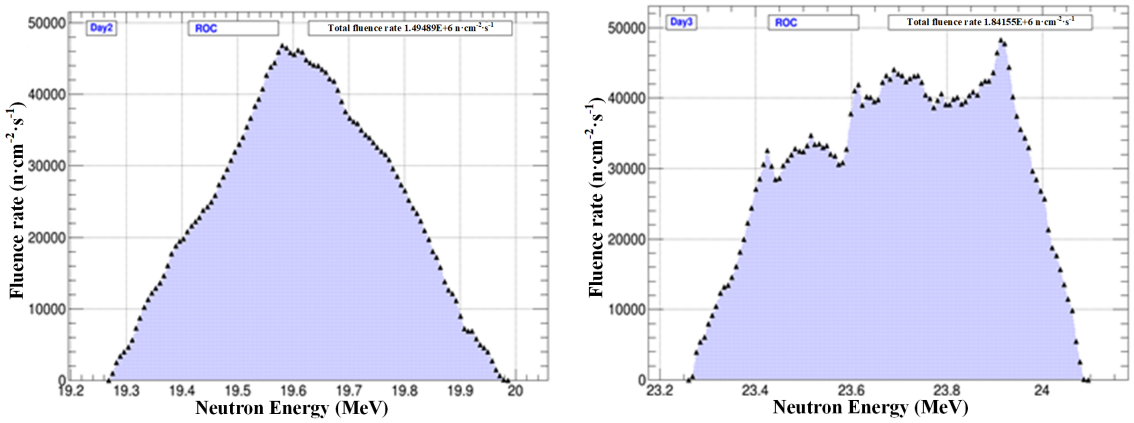
Table 4.4: The recorded FF SEUs per beam run, the confidence intervals and the associated cross-sections computed after the methodology from [38], considering a fluence uncertainty of ±10% and a 95% confidence level.

Neutron energy [MeV]	FF SEUs	FF SEUs lower	FF SEUs upper	σ_{FF} [$\times 10^{-14}$ cm^2·bit^{-1}]
20 (runs 1, 2, 5 - 7)	22	13.79	33.31	$1.37^{+0.75}_{-0.58}$
22 (run 4)	10	4.80	18.39	$3.46^{+2.98}_{-1.93}$
24 (runs 3, 8 and 9)	37	26.05	51.00	$6.28^{+2.68}_{-2.23}$

Table 4.5: The recorded FF SEUs per beam energy, the confidence intervals and the associated cross-sections computed after the methodology from [38], considering a fluence uncertainty of ±10% and a 95% confidence level.

and 4 MeV. In [193] it is also mentioned that parasitic neutrons are expected between 1 and 5 MeV. Both papers present experimental results of the applied multiple foil activation technique. In particular, the ultra-low energy parasitic neutrons can induce (n, γ) reactions with a very high probability, producing β-unstable nuclei, such as ^{31}Si, which in turn yield highly ionizing electrons, up to 7 MeV, during their de-excitation. In this energy regime, all the above-mentioned processes are not well studied and accordingly, there is a high degree of uncertainty related to their contribution. Currently, there are no published measurements of the parasitic neutrons when the nominal energy is 20, 22 or 24 MeV for this setup. Together with the increased effect of the energy, this is a possible explanation for the increased cross-section at 24 MeV with regards to 20 MeV. Ideally, these effects should be studied on the ATLAS experimental setup to conclude for the corresponding effective cross-sections and convolute for the expected LHC neutron spectrum. Nonetheless, these effective cross-sections are a good approximation.

The recorded error counter values and the captured ILA waveform windows were analyzed at the end of each run. For all nine runs, no misalignment, encoding, decoding or protocol syntax errors were detected for any of the four ROC output data streams proving that the ROC logic was not driven into illegal states and



(a) (b)

Figure 4.7: Ultrafast neutron flux spectra at the ROC distance of 1.27 cm, when the beam energy is (a) 20 MeV (run 1) and (b) 24 MeV (run 3) (provided by the accelerator operators from NCSR) [174].

recovered gracefully from the FF SEUs. For all runs except the third and the last, the SROC output packet counters had values close to each other as they should meaning that no de-synchronization between the L1 triggers and the input events happened in any SROC. All four SROCs should output the same number of L1 packets since they respond to the same L1 triggers but their size varies in time and from one SROC to another. Thus, the outputs are not perfectly synchronized even if the transmission speed is identical and one cannot expect the output packet counters to always have identical values, but close to each other. In Table 4.6, the counts for the encountered errors in the output packets, classified by error type (see Section 4.2.2) are shown. For runs 3 and 9, single instances of the unexpected Busy-On/Off type of error were observed. The Busy-On/Off error type is linked to the loss of synchronization between the L1 trigger and the L0 data and is detailed at the end of this section. Using the ILA-produced waveform files, all the encountered error types have been analyzed. For all of them, the root cause was determined to be bit flips in the data-occupied SRAM latches. The expected error types order, from the most to the least frequent, was correct. As with the FF SEUs, the number of output packet errors encountered during run 3 (24 MeV) was several times larger than run 1 and 2 (both 20 MeV), despite similar conditions except the nominal beam energy. Also, as a reference, the ROC sent 7.78×10^{11} packets during the nine beam runs and the total number of output packet errors was 5257. Even considering that each of these errors was signaled for a different L1 packet, the resulting proportion of affected packets is small (one in 147,993,152).

In Figure 4.8 ILA captured data is depicted as waveforms showing an example of an L1 hit word that raises the hit content and parity error flags and a trailer with incorrect checksum. The altered L1 packet originates from SROC3. The first three signals from the top represent the 8b10b decoded SROC data stream (i.e. the decoded byte, the 1-bit signal indicating if the byte is normal data or a special symbol and the 1-bit signal that validates these data. The next two signals are the SROC assembler data output and its validation signal (see Chapter 3, Section 3.2.2). The format is the one presented in Figure 2.16 from Chapter 2, Section 2.3.2. The rest of the signals generate pulses for the different types of error flags,

Error \ Run	1	2	3	4	5	6	7	8	9	Total
hit content	367	199	1248	214	80	27	3	67	435	2640
parity	236	152	907	155	59	17	2	51	333	1912
VMM missing	109	35	287	45	24	5	1	10	134	650
checksum	4	4	23	6	1	2	0	0	3	43
Orb. & BCID	0	0	1	2	0	0	0	0	1	4
VMM	0	1	1	0	0	0	0	0	0	2
L1ID	0	1	0	1	0	0	0	0	0	2
Busy ON/OFF	0	0	1	0	0	0	0	0	1	2
No. hits	0	0	1	0	0	0	0	0	0	1
ROC ID	0	0	0	0	0	1	0	0	0	1
Total	716	392	2469	423	164	52	6	128	907	5257

Table 4.6: Per run counts of the packet errors caused by SEUs occurring in the SRAM FIFO buffers.

except for *state* which represents the internal state of the SROC Assembler FSM and is used for debugging flow errors that, as expected, were not observed during the radiation tests. The signals corresponding to the three types of observed errors are highlighted. The moment when the checksum error is detected is emphasized with a vertical marker. By inspecting the entire packet, it was determined that a bit flip from 1 to 0 of the *rel_BCID* field's MSB within the 8th hit word (i.e. changing its value from 2 to 0) caused hit content and parity errors for that hit. Since the flip happened in the SROC packet FIFO the checksum of the packet indicates that the packet was altered. By reverting the bit flip, the packet becomes completely valid. For all the acquired ILA windows, similar analyses were performed.

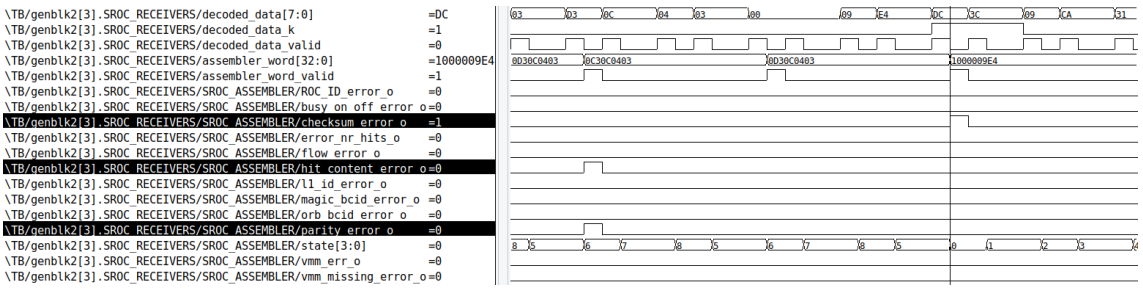


Figure 4.8: Example ILA waveforms depicting checksum, hit content and parity errors caused by a bit flip within a hit word from an L1 packet.

In Figures 4.9 and 4.10, the arrivals in time of FF SEUs and output packet errors are depicted for a window of 700 iterations (i.e. 7000 s since 1 iteration = 10 s) during run 1 (20 MeV) and 3 (24 MeV), respectively. These time windows represent subsets of the beam run-times mentioned in Table 4.3. Hence, the depicted FF SEUs and output packet errors are also subsets of what is reported in Tables 4.4 and 4.6. The same time window size is used in both figures to visually depict the

relative rates of errors in the same conditions, except for the beam energy. The output packet errors are classified by type (see Subsection 4.2.2) and are listed in descending order starting from the most frequent at the bottom to the least frequent just below the FF SEUs trace. The packet error types are distinguished by color. Multiple errors of the same type can be produced within one iteration, especially because the packets from all four SROCs are checked in parallel. Thus, for each error type, the vertical position of the marking point indicates the number of new encountered errors in that iteration. Relatively high spikes for the *VMM missing* errors were observed, as shown in both figures. They are caused by the loss of synchronization between the L1 triggers and the L0 packets. The ten new *VMM missing* errors shown in Figure 4.10, after iteration 600, were recorded in the same iteration as the Busy-On/Off error. The counts from Table 4.6 show that this type of error is the third most frequent, after the hit-content and parity errors even though these express smaller spikes.

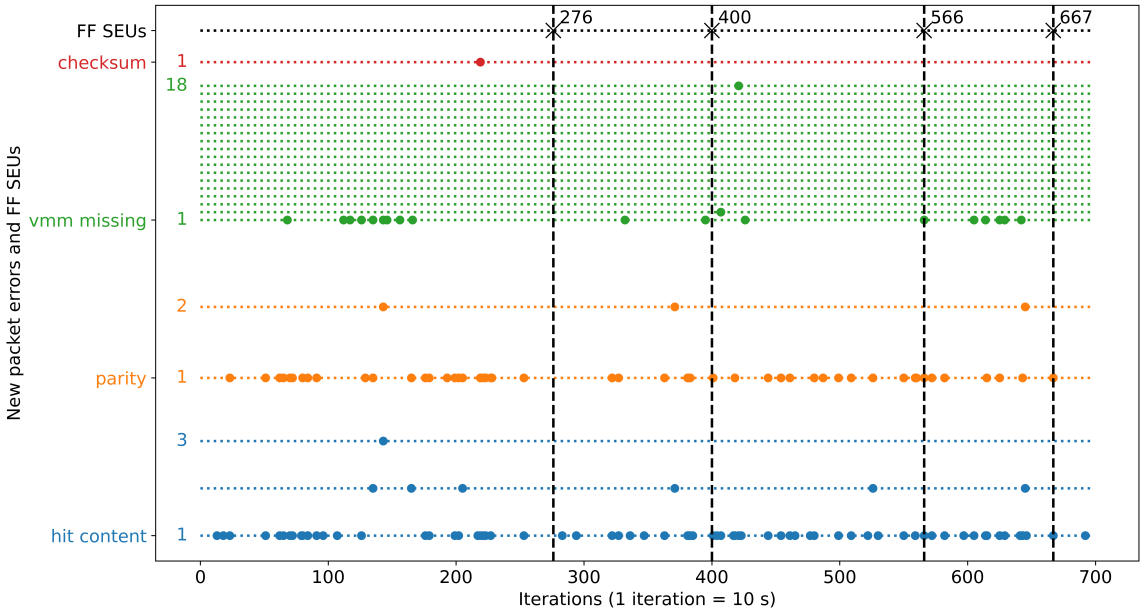


Figure 4.9: A time window of 700 consecutive iterations from run 1 (3816 complete iterations) depicting with markers the moments of detection for four of the FF SEUs (out of the total 11 in run 1) and various output packet errors separated by type.

In Figures 4.11 and 4.12, the normalized histograms of the interarrival times for the FF SEUs and the first three most frequent types of output packet errors are depicted for a subset of run 1 (20 MeV) and the entire run 3 (24 MeV), respectively. During run 1 the test bench was restarted several times for debugging purposes but the chosen subset represents the largest uninterrupted interval. The bins are of equal size and cover the entire interval between the minimum and maximum encountered interarrival times. The histogramming time resolution is determined by the interarrival times which are only integer multiples of one iteration (i.e. 10 s). The multiple events of a certain type that were recorded within one iteration are not considered for the leftmost bin. Just the differences between the iterations with events of the same type are used. Therefore, the depicted mean arrival rates per iteration (i.e. λ) are not the total average rates of occurrence. The ideal exponential probability density function has the same λ rate as the samples since a steady-state

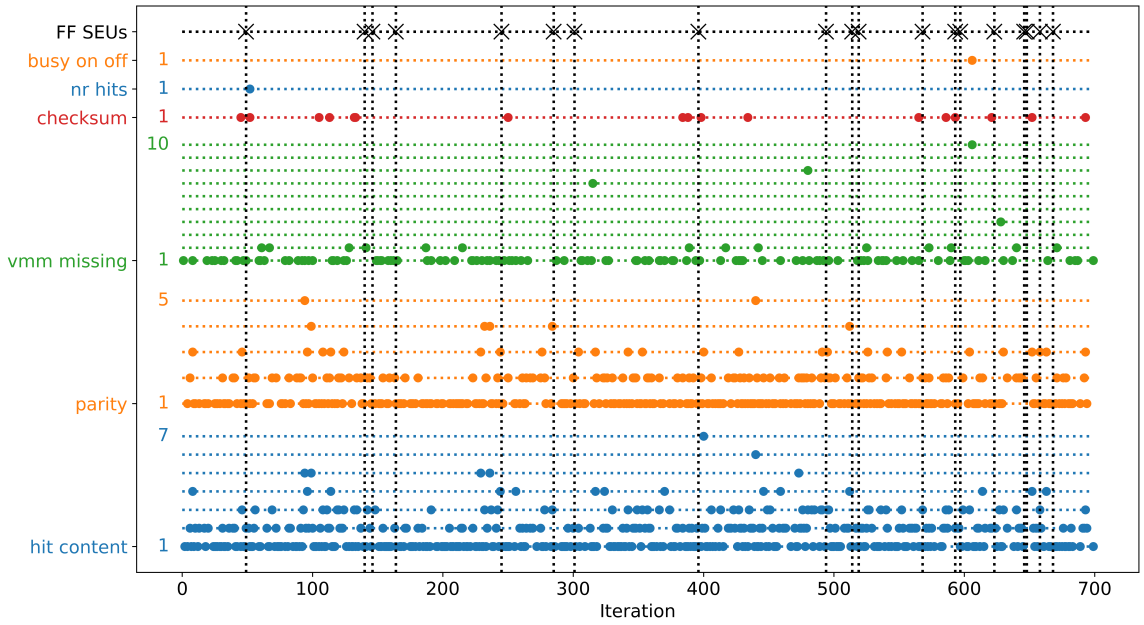


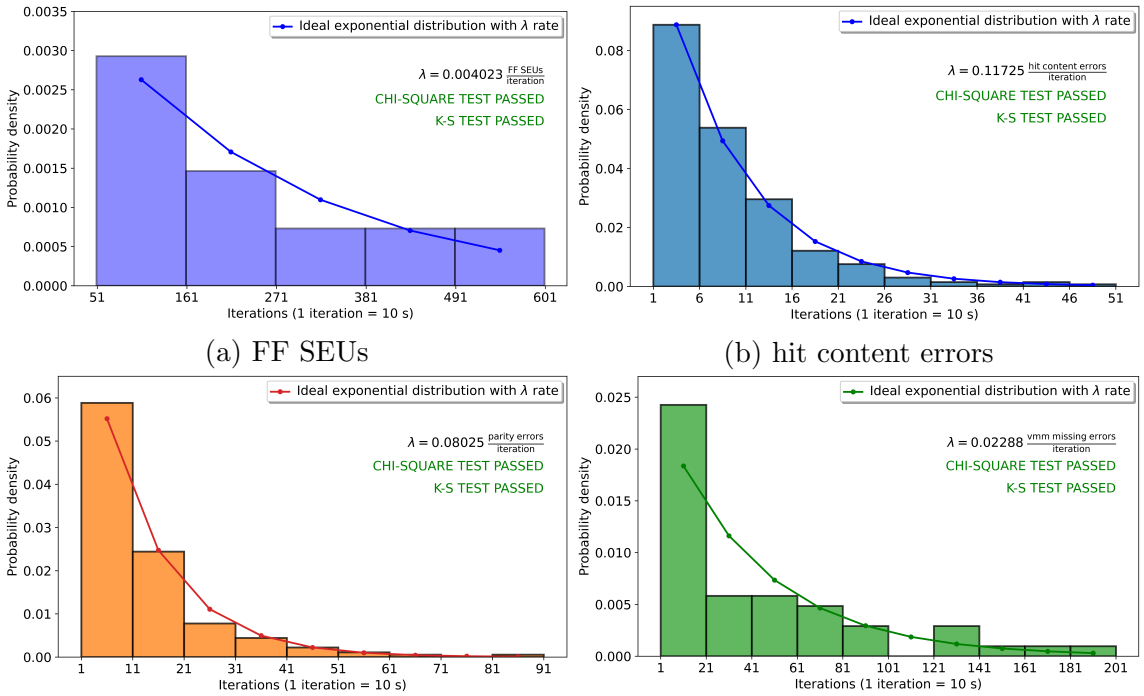
Figure 4.10: A time window of 700 consecutive iterations from run 3 (1224 complete iterations) depicting with markers the moments of detection for 19 of the FF SEUs (out of the total 31 in this run) and various output packet errors separated by type.

process is assumed. The probability that the interarrival time is above the right edge of the rightmost bin is not displayed but it is taken into account in the scaling of both the theoretical and empirical densities. This also applies to the probability that the interarrival time is less than the left edge of the leftmost bin in the case of FF SEUs. To analyze the validity of the memoryless hypothesis, goodness of fit tests were performed in all cases. Their verdicts are listed in each histogram. The following paragraphs detail these assessments.

In Table 4.7 the results of the Chi-square and Kolmogorov-Smirnov goodness of fit assessments are listed. For the Chi-square the same bins as in the histograms were used. The number of Degrees of Freedom (DoF) equals the number of bins minus the two constraints: i) the same theoretical mean arrival rate as in the observed data and ii) the same theoretical total population size as the observed one. The χ^2 value is computed as in Equation 4.1, where f_o and f_e are the observed and expected frequencies, respectively. For the Kolmogorov-Smirnov analysis, the maximum absolute error between the empirical and the theoretical cumulative distribution functions (noted as D_n where n represents the sample count) was computed. The critical values correspond to a 5% significance level. A test is considered passed if the resulting value is smaller than the critical value. In Figures 4.13 and 4.14, graphical representations of the differences between the empirical and the ideal values for both assessments in all these cases are presented.

$$\chi^2 = \sum_{i=1}^{DoF+2} \frac{(f_o - f_e)^2}{f_e} \tag{4.1}$$

The results show that it is correct to assume exponential distributions for both the FF and the SRAM SEUs at 20 MeV (run 1). At 24 MeV (run 3), the memoryless characteristic is confirmed only for the FF SEUs and the *VMM missing* errors. The



(a) FF SEUs (b) hit content errors

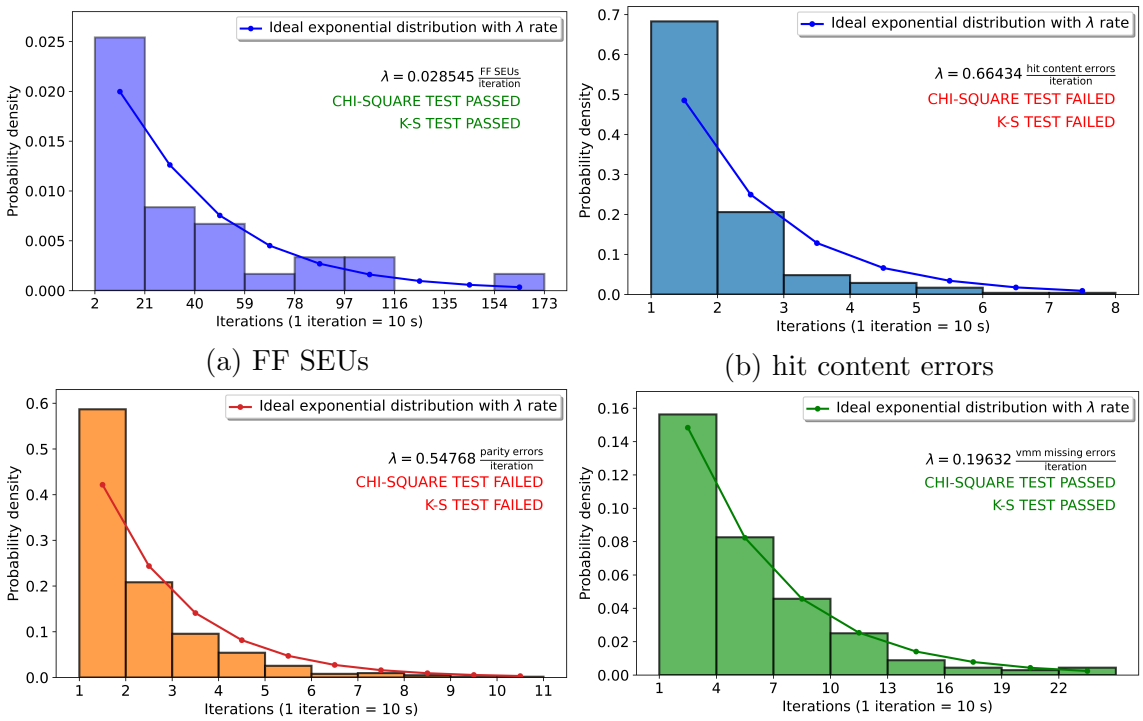
(c) parity errors (d) *VMM missing* errors

Figure 4.11: Normalized histograms and theoretical probability density functions of the interarrival times in case of the FF SEUs and the three most frequent types of output packet errors observed within an uninterrupted window of 2250 iterations from run 1 (20 MeV, 3816 iterations).

Error	**Test**	**Chi-square**				**Kolmogorov Smirnov**			
	Run	χ^2	***Dof***	**crit. val.**	**verdict**	D_n	n	**crit. val.**	**verdict**
FF SEU	1	0.41	3	7.81	Pass	0.11	9	0.43	Pass
	3	5.13	7	14.07	Pass	0.12	31	0.22	Pass
hit content	1	4.67	8	15.51	Pass	0.04	263	0.08	Pass
	3	137.98	5	11.07	Fail	0.2	760	0.05	Fail
parity	1	7.09	7	14.07	Pass	0.05	180	0.10	Pass
	3	78.50	8	15.51	Fail	0.17	626	0.05	Fail
VMM missing	1	11.86	8	15.51	Pass	0.18	51	0.19	Pass
	3	4.01	6	12.59	Pass	0.05	224	0.09	Pass

Table 4.7: Details of the Chi-square and Kolmogorov Smirnov goodness of fit assessments of the interarrival times from Figures 4.11 and 4.12, relative to the ideal exponential probability density functions with the same λ (rate).

used time resolution is not satisfactory for the *hit content* and *parity* errors. Since one SRAM bit flip can cause the detection of multiple packet errors, one cannot assume exponential distributions for the interarrival times of the output packet errors even if the SRAM SEUs that caused them do follow such distributions. This



(a) FF SEUs

(b) hit content errors

(c) parity errors

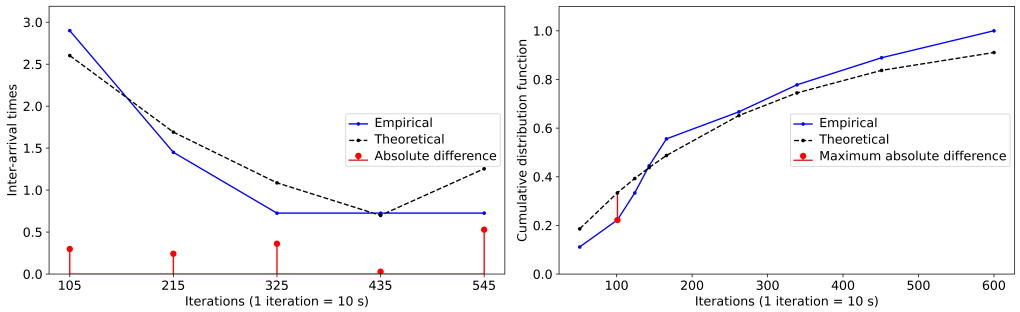
(d) *VMM missing* errors

Figure 4.12: Normalized histograms and theoretical probability density functions of the interarrival times in case of the FF SEUs and the three most frequent types of output packet errors observed within the entire uninterrupted 1224 iterations of run 3 (24 MeV).

effect can be visualized in the failing 24 MeV histograms where the value of the first bin is significantly larger than the estimation and is the main cause of failure for the goodness of fit tests. However, the memoryless hypothesis is confirmed for the FF SEUs case.

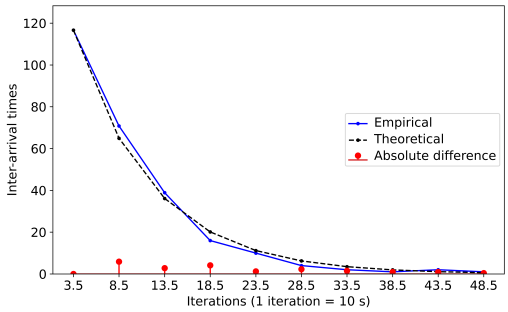
The χ^2 test is sensitive to the choice of bin edges. Also, if all the encountered FF SEUs and output packet errors are considered, both the χ^2 and Kolmogorov-Smirnov goodness of fit results change since λ (i.e. average rate) changes. If the bins are kept the same except for of the rightmost and leftmost ones, the results change. The rightmost bin covers now up to infinity while the leftmost one starts in all cases with the one interation and below interarrival time. Thus the multiple errors encountered within an iteration are accounted for. The goodness of fit assessments in these conditions are detailed in Table 4.8.

As described in Section 4.2.2, one bit flip can cause multiple errors. Therefore, the error counts from Table 4.6 are an overestimation of the SEUs affecting the SRAM latches. Nevertheless, they can be used to approximate the probability of SRAM SEUs. The number of occupied SRAM bits that could cause each error type, if altered, was estimated based on the distributions of occupied SRAM bits (see Table 4.2) and the descriptions from Section 4.2.2. The calculations are detailed in Table 4.9 for the four most frequent types of errors. The estimations do not take into account the cases when the headers, hit or trailer words are mistaken for each other due to alterations of the additional bits which signal the end of packets (i.e. the bits with index 32 in Figures 2.14 and 2.16). In these cases, there is a high

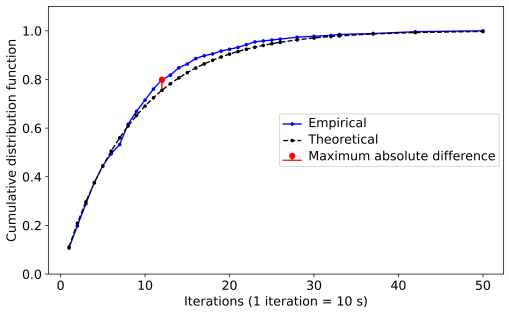


(a) χ^2 for FF SEUs.

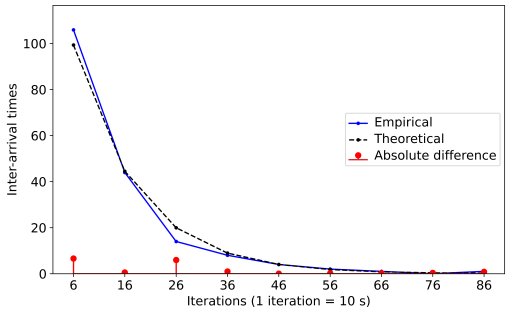
(b) K-S for FF SEUs.



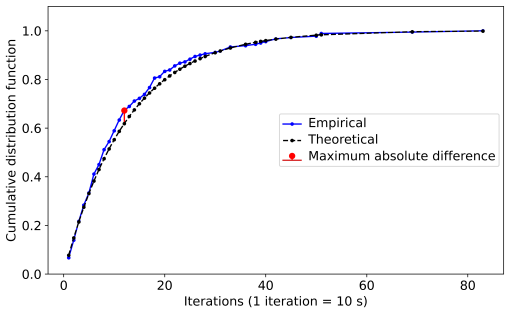
(c) χ^2 for hit content errors.



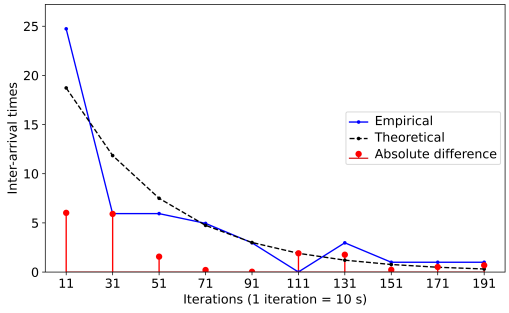
(d) K-S for hit content errors.



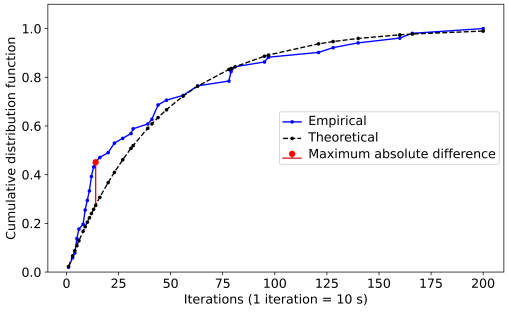
(e) χ^2 for parity errors.



(f) K-S for parity errors.



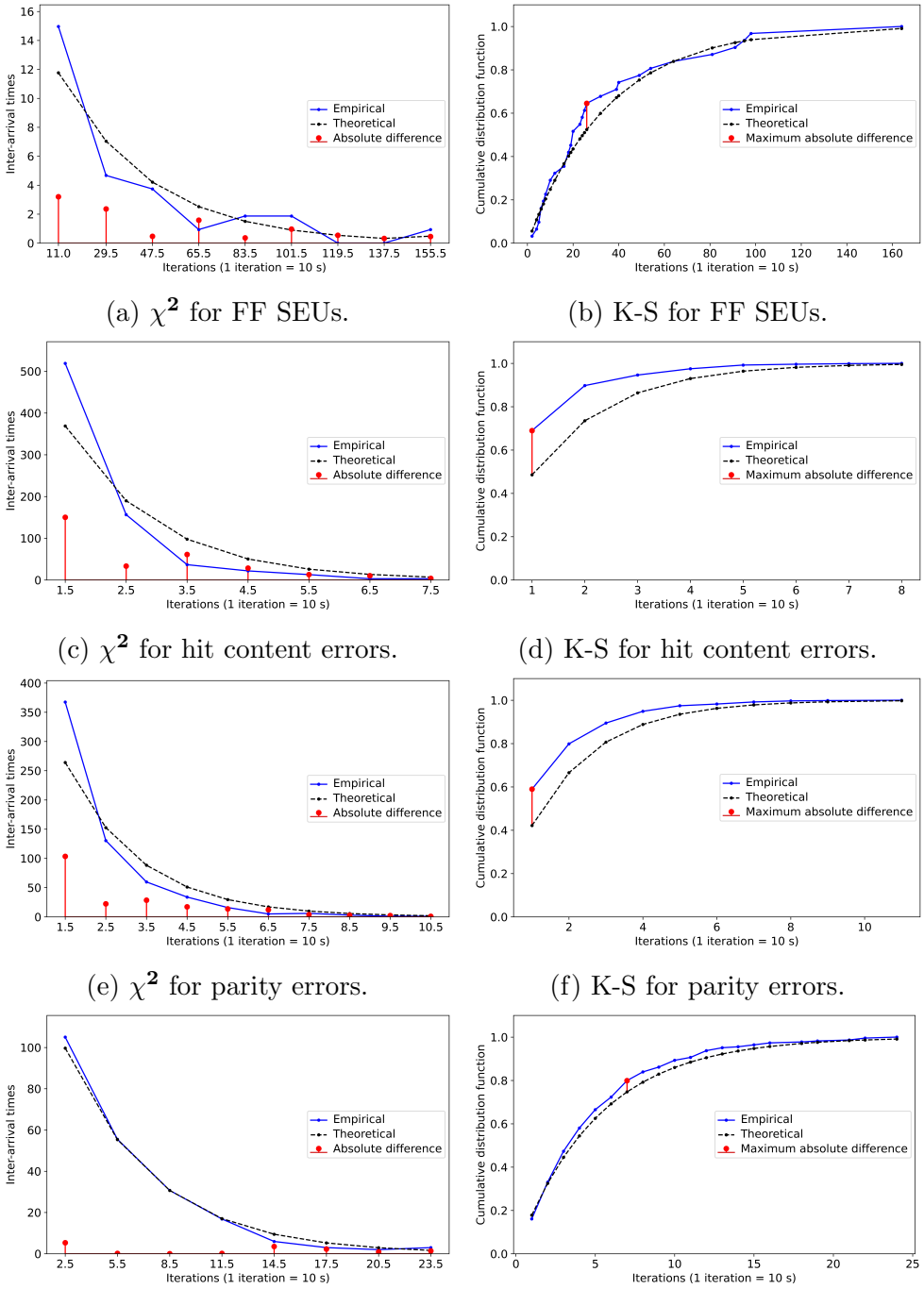
(g) χ^2 for *VMM missing* errors.



(h) K-S for *VMM missing* errors.

Figure 4.13: The absolute differences between the theoretical and the empirical histograms of interarrival times (a, c, e and g) and the maximum absolute errors between the empirical and theoretical cumulative distribution functions (b, d, f and h) for run 1.

probability that the apparent parity bits or checksum fields will be incorrect. It was considered that these cases are negligible relative to the other error causes and thus they do not change the values from Table 4.9 by a significant margin. Also, for the *VMM missing* type of error, the scenario when at least one VMM3 Capture FIFO overflowed because the associated SROC was searching for altered L1 triggers



(a) χ^2 for FF SEUs.

(b) K-S for FF SEUs.

(c) χ^2 for hit content errors.

(d) K-S for hit content errors.

(e) χ^2 for parity errors.

(f) K-S for parity errors.

(g) χ^2 for *VMM missing* errors.

(h) K-S for *VMM missing* errors.

Figure 4.14: The absolute differences between the theoretical and the empirical histograms of interarrival times (a, c, e and g) and the maximum absolute errors between the empirical and theoretical cumulative distribution functions (b, d, f and h) for run 3.

was not considered. In all these uncovered cases the outcome depends on the data content, which varies in time for the triggers, headers and trailers words. Thus, one cannot distinguish the exact cause of an error based solely on the output packets.

The analysis could be taken further by estimating the bit flips of each type of FIFO using the details presented in Section 4.2.2. However, in many cases, it is

Error	Test	Chi-square				Kolmogorov Smirnov			
	Run	χ^2	*Dof*	crit. val.	verdict	D_n	n	crit. val.	verdict
FF SEU	1	0.26	3	7.81	Pass	0.11	9	0.43	Pass
	3	4.91	7	14.07	Pass	0.12	31	0.22	Pass
hit content	1	4.30	8	15.51	Pass	0.09	299	0.08	Fail
	3	131.01	6	12.59	Fail	0.15	1248	0.04	Fail
parity	1	5.88	7	14.07	Pass	0.05	190	0.10	Pass
	3	112.75	9	16.92	Fail	0.17	907	0.05	Fail
VMM missing	1	51.70	8	15.51	Fail	0.37	85	0.15	Fail
	3	7.59	6	12.59	Pass	0.12	287	0.08	Fail

Table 4.8: Details of the Chi-square and Kolmogorov Smirnov goodness of fit assessments of the interarrival times for runs 1 and 3, relative to the ideal exponential probability density functions with the same λ (rate), considering all the recorded events and the rightmost and leftmost bins are extended.

Error type	Number of corresponding bits occupied in SRAM
hit content	VMM Capture and SROC FIFOs hit data: $8 \times 980.83 \times 28 + 4 \times 18.56 \times 28 = 221784.23$ bits
parity	VMM Capture and SROC FIFOs hit data covered by the parity bit and SROC FIFO BCIDs: $8 \times 980.83 \times 23 + 4 \times (18.56 \times 23 + 1.13 \times 12) = 182234.38$ bits
VMM missing	From VMM Capture FIFO: L0 headers (OrbitID, BCID, overflow flag, parity bit) and hit data (truncate flag); from SROC FIFO: *VMM missing* field from trailer words; from BC FIFO, all L1 triggers (OrbitID and BCID): $8 \times (246.6 \times 16 + 980.83) + 4 \times (1.13 \times 8 + 0.22 \times 12) = 39458.12$ bits
checksum	All data from SROC FIFO: $4 \times 694.19 = 2776.76$ bits

Table 4.9: Estimations of the number of occupied SRAM bits that could cause each of the four most frequent output packet errors, if altered.

impossible to determine exactly where the bit flip happened. Significantly more bit flips were expected to happen in the VMM Capture FIFOs compared to the SROC packet FIFOs, even though they use the same macro buffer design, because their occupancy levels are significantly larger, as shown in Section 4.2.2. Also, from this point of view, even fewer bit flips were expected in the TTC FIFOs. Therefore, the same cross-section is considered for the three functional types of SRAM.

The counts for the first four most frequent types of output packet errors, over the runs with the same beam energy, are listed in Table 4.10. In Figure 4.15 the cross-section confidence intervals for the FF SEUs and these four most frequent types of packet errors are depicted categorized by the energy of the neutron beam that produced them. The values for the FF SEUs are the ones from Table 4.5, while the values for the packet errors are based upon Table 4.10. They assume a 10% fluence

uncertainty and use a 95% confidence level. The results show that the probability for the SRAM latches to suffer SEUs is approximately equal between them. As shown in Figures 4.9 and 4.10, the rate of packet errors is significantly increased at 24 MeV compared to 20 MeV, translating into the relatively substantial cross-section values from Figure 4.15. This can be explained by the more frequent loss of synchronization between the L1 triggers and the L0 data. The desynchronization causes important data loss and the sending of sparse packets which trigger more errors. In extreme cases, the SROC is not able to recover from these situations, requiring a reset. Such particular events occurred during runs 3 and 9, both with 24 MeV neutron beam energy, as the SROC 3 remained far behind the other three in the number of transmitted packets. At the same time, single Busy-On/Off errors were also recorded from SROC 3. To recover, SR pulses were issued through the TTC stream. An analysis of the link between these extreme losses of synchronization and the Busy-On/Off errors was performed. In the following paragraphs, this detailed investigation is presented.

Neutron energy [MeV]	**hit content**	**parity**	***VMM missing***	**checksum**
20 (runs 1, 2, 5, 6 and 7)	676	466	174	11
22 (run 4)	214	155	45	6
24 (runs 3, 8 and 9)	1750	1291	431	26
Total	**2640**	**1912**	**650**	**43**

Table 4.10: The counts of the four most frequent types of recorded packet errors, caused by SRAM SEUs, per beam energy.

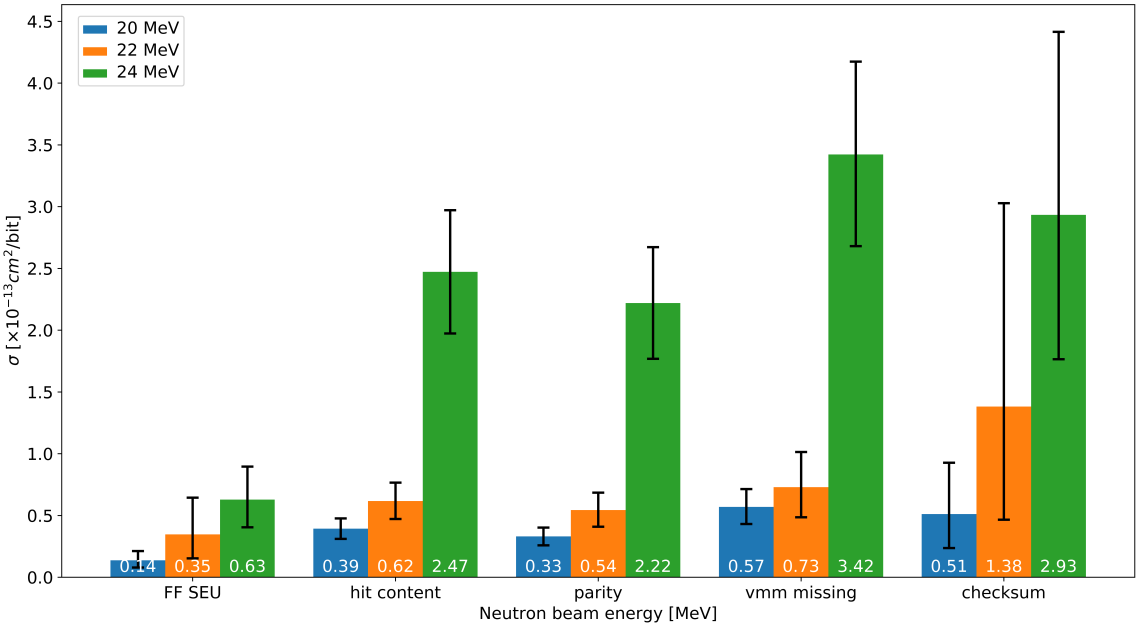


Figure 4.15: The computed cross-sections of the FF SEUs and the four most frequent types of packet errors, by neutron beam nominal energy, considering a fluence uncertainty of 10% and a 95% confidence level.

If the Busy-On/Off mechanism is enabled (as in the used default ROC configuration), special 8b10b flow control symbols (i.e. Busy-On and Busy-Off) are sent between the SROC output packets to signal when the occupancy level of any associated VMM Capture FIFO passes above a configurable threshold or below a second lower configurable threshold (see Chapter 2, Subsection 2.3). These symbols are not buffered into the SROC Packet FIFOs but inserted by the Streamer FSM alongside SOPs, EOPs and *commas*. The Busy-On/Off error is generated if the sequence of these symbols is not alternative or does not start with *Busy-On*. By default, the *Busy-On* threshold is 1900 and the *Busy-Off* limit is set to 1800. Both values are larger than the average VMM Capture FIFO occupancy of 1227.43 with an amount that is not reached by the variation of this occupancy level, as shown in Figure 4.6. Thus, it was not expected to see any *Busy-On* or *Busy-Off* symbols in the output SROC streams during the ROC irradiation testing. However, the loss of synchronization between the L1 triggers and the L0 data produces large variations of the occupancy levels. *Busy-On* or *Busy-Off* symbols could have been sent causing the rise of the unexpected error.

Bit flips in the associated TTC FIFO might have caused the SROC to search for incorrect BCs. Besides, bit flips in the associated VMM Capture FIFOs could cause a header to be mistaken as a hit word or vice versa, even with correct parity. The SROC will lose the synchronization with the VMM data causing the VMM Capture FIFO to go full (when the data appears to be newer than the searched trigger) and then empty (when the data appears to be older than the searched trigger). Consequently, *Busy-On* and *Busy-Off* symbols might have been sent. Various other errors would be reported because input data and L1 triggers are discarded (e.g. *VMM missing*). This situation becomes critical when the SROC cannot recover by itself and loses all further data. In this case, an SR command or a pulse on the asynchronous reset should be issued and the chip re-initialized. This behavior could have been avoided by implementing TMR for the bits marking the end of packets inside the VMM Capture FIFOs. The triggers could have been protected by implementing TMR for the orbit and BCID fields or by adding ECCs. However, these techniques were not mentioned in the requirements document and would have required larger SRAM memories and more associated logic. To minimize data loss, FELIX should monitor the number of output packets from each SROC. An SR command should be issued to the ROC chip whose SROC did not transmit any hit-data in a configurable period and remained behind the rest. The situation will also be signaled by multiple consecutive packets with set *VMM missing* flags and unexpected BC information. The associated hit data should be discarded even if the checksum fields and the parity bits are correct.

In Figure 4.16 the evolution of the occupancy levels for all the FIFOs associated with the third SROC are depicted, in a simulation with the same parameters as the irradiation test, when a single bit flip in the TTC FIFO changes the triggered orbit from *0b10* to *0b00*. By design, the SROC logic considers that the VMM3 data are newer than the L1 trigger in two situations (see Chapter 2 Section 2.5): i) if the searched and the L0 OrbitID are equal and the searched BCID is smaller than the one in the L0 header and ii) if the searched OrbitID is behind the L0 one with one increment. An output packet is formed only when the entire BC information matches. In all other cases the VMM3 data are considered older than the trigger and the SROC logic will discard it trying to catch up with the trigger. This happens

in the presented case for VMM Capture 3 until the L0 OrbitID becomes *0b00* as the L1 trigger (i.e. passing through the *0b10* and *0b11* values) and the L0 BCID becomes larger than the triggered BCID. Since the discarding of this data happens faster than packets are being received, the fill level falls. During this, VMM Capture 4 only receives data and its fill level rises. When the VMM Capture 3 data become newer than the altered L1 trigger, the SROC begins the same process with VMM Capture 4. The VMM Capture 3 fill level begins to rise again as data are not read. During this, the other L1 triggers queue up in the TTC FIFO. When the VMM Capture 4 data become newer than the altered trigger, a header - trailer packet is generated in response to the altered trigger. The packet will have (incorrect) hit data only if the BCID of the altered trigger matches an L0 packet with hit data in that OrbitID. For the next unaltered L1 triggers data will appear to be newer and the SROC will output header and trailer only packets (no hit words) with the *VMM missing* flags set for both VMM Captures. Starting with the first trigger newer than the value of the altered one, the ROC will be recovered and the output data will be correct. This was checked in a simulation in which the hit content varies to be sure that the hit words belong to the packet they are supposed to. The common traces with Figure 4.6 have the same color. In this case, the hit data of two OrbitID are lost ($2 \times 89.1\ \mu s \times 1.37\ \text{MHz} \approx 244$ packets). Since there are two VMM Capture associated with the SROC, it means that the hit data of 488 input packets are lost. The maximum fill level for VMM Capture 4 is not enough to cause the sending of a *Busy-On*. Without other SEUs the SROC can recover by itself.

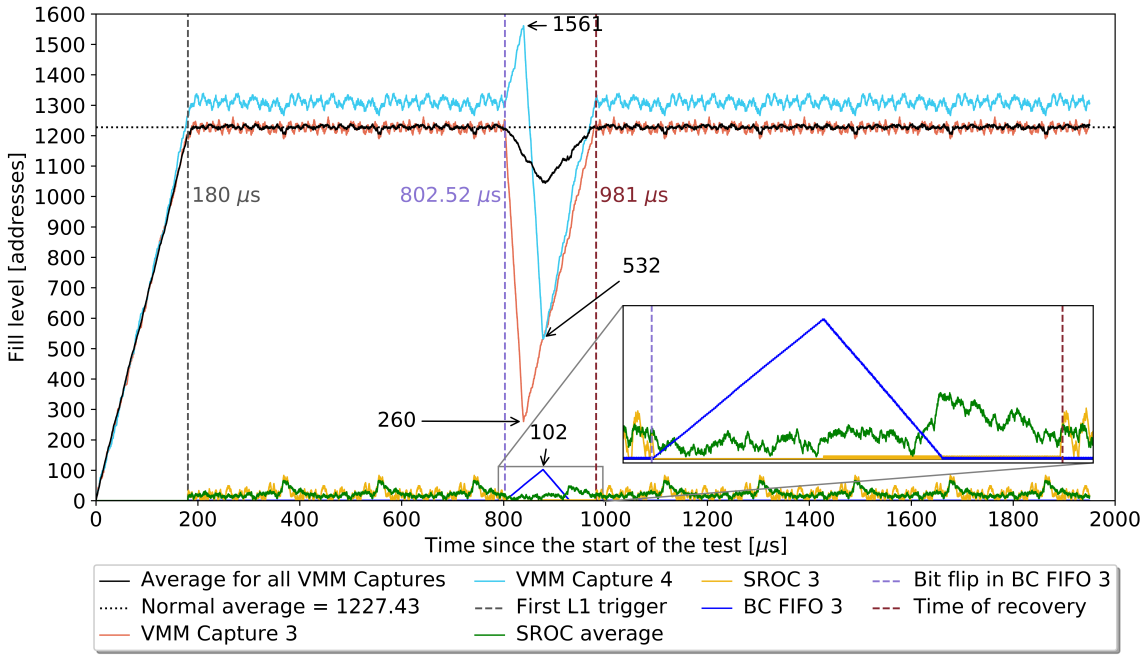


Figure 4.16: The evolution of the fill levels for the FIFOs associated with SROC 3 when the OrbitID of an L1 trigger buffered within the TTC FIFO suffers a bit flip from 1 to 0.

In Figure 4.17 the same traces are depicted in the same scenario with the exception that an additional bit flip within the same TTC FIFO alters another L1 trigger, relatively close to the initial altered one. This could be motivated by the increased processing time of the initial altered L1 trigger causing the following triggers to

accumulate and wait longer than normal within the TTC FIFO and thus having an increased probability of being corrupted through SEUs. In this case, the SROC does not have enough time to recover before the second altered trigger. Busy-On and Busy-Off symbols are issued as the fill levels surpass the corresponding thresholds and consequently the Busy-On/Off error is raised. In this case, the SROC loses all further data and the logic must be reset. Figures 4.6, 4.16 and 4.17 and their interpretations were obtained from the analysis of RTL simulations in which the ROC and the test environment were configured identically as during the irradiation testing. This allowed better controllability and observability than in the real test setup. It is considered that during the irradiation testing, SROC 3 suffered two desynchronizations similar to the one from Figure 4.17 causing the two Busy-On/Off errors. This behavior is not specific to SROC 3. In the used configuration, the hit words buffered by the associated VMM Capture 3 and 4 channels could be confused more easily with a valid hit header.

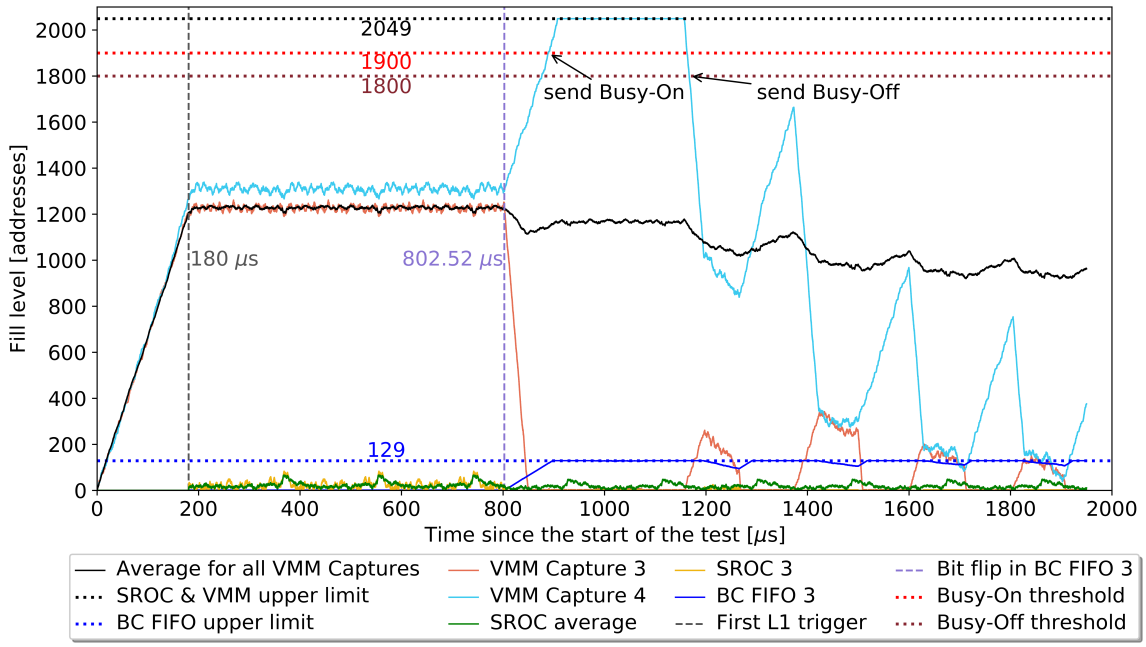


Figure 4.17: The evolution for the FIFO fill levels when two relatively close L1 triggers suffer bit flips while being buffered within TTC FIFO.

4.3 Conclusion of the irradiation tests

The ROC is an essential on-detector real-time packet processor ASIC for the NSW read-out system. For assuring its immunity to radiation-induced events, TMR with independent clock trees was implemented for its FFs. A significant area of the silicon die is occupied by off-the-shelf SRAMs without a radiation-hardened design, implemented ECCs or TMR but the data stored in these SRAMs have checksum data fields and parity bits. The mass-production test setup, presented in Chapter 3, was adapted and used to assess ROC's behavior in an ultrafast neutron environment. The ROC context was emulated using a Xilinx FPGA evaluation board interfaced with custom PCBs accommodating the chip samples. The Tandem accelerator from

the NCSR Demokritos, Athens, Greece produced the necessary ultrafast neutron beams for the irradiation tests.

Two ROC chip samples were subjected to neutron beams with energies of 20, 22 and 24 MeV for 30, 2 and 7 hours, respectively. The test configuration stressed the ROC with rates and latency above the maximum expected ones. A total of 69 SEUs were reported in the 18,458 FFs of the ROC digital part while the die was hit by $3 \times 10^{10} \pm 10\%$ neutrons. This quantity is the equivalent of 8 months of LHC operation at luminosity $L = 10^{34}$ p/cm^2/s in the most irradiated part of the NSW detector. More than half of the FF SEUs happened during the 24 MeV beams. The total average FF cross-section is $2.78^{+0.92}_{-0.82} \times 10^{-14}$ cm^2/bit. The average per beam energy FF cross-sections are $1.37^{+0.75}_{-0.58} \times 10^{-14}$; $3.46^{+2.98}_{-1.93} \times 10^{-14}$ and $6.28^{+2.68}_{-2.23} \times 10^{-14}$ cm^2/bit at 20, 22 and 24 MeV energies, respectively. The proper ROC operation was not affected by these FF SEUs. Several output packets were however corrupted by SEUs produced within the SRAM latches. The test setup classified and counted the errors and captured the relevant waveforms of the affected packets. One of the worst-case effects of a single SRAM bit flip is presented. Multiple close SRAM bit flips could cause the loss of synchronization between the VMM3 data and the L1 triggers. Such a scenario was simulated and detailed. During the irradiation testing, two similar events caused important data loss in one of the SROCs. The chip requires a reset to recover from this state. A suggestion for minimizing the data loss, in this case, is presented.

The SRAM SEUs cross-sections were estimated at $5.82^{+1.18}_{-1.18} \times 10^{-14}$; $9.16^{+2.01}_{-1.99} \times 10^{-14}$ and $37.18^{+7.39}_{-7.39} \times 10^{-14}$ cm^2/bit at 20, 22 and 24 MeV energies, respectively. These values are more pessimistic than in reality because they are based upon the total output packet error counts, but a single bit flip may cause multiple errors to be reported. All the stated confidence intervals use a 95% confidence level and take into account the 10% fluence uncertainty. Considering the obtained 24 MeV cross-sections, the estimation is that each ROC will experience $150 < 232 < 331$ FF and $17593 < 21957 < 26323$ SRAM SEUs per year at $L = 10^{34}$ p/cm^2/s. These amounts are equivalent to $1.39 < 2.15 < 3.06$ FF SEUs/min and $163.07 < 203.52 < 243.99$ SRAM SEUs/min, respectively, in all 4875 ROC ASICs which will be installed for NSW. One can extrapolate the values for the next LHC runs when the luminosity will first double and then reach five times the luminosity used as reference.

Chapter 5

An application

During the development of the FPGA design for the validation of the ROC digital design, its digital functional mass-testing and irradiation qualification, one of the most useful deployed tools was the Xilinx IP ILA [206]. It increased considerably the observability within the FPGA design and therefore its debuggability and allowed the extensive analysis of the recorded radiation-induced output packet errors as presented in Chapter 4, Section 4.2.3. However, the Xilinx FPGA ILA has some constraints and limitations. Thus, the idea of a new FPGA ILA design that mitigates these limitations but introduces new ones was born. It proposes changes to both the current method and apparatus which improve the performance and efficiency of the tool, but of course, increases its complexity.

In section 5.1, the generic Logic Analyzer (LA) tool is presented. Its main components are detailed and their roles emphasized. Different LA types are presented. In Section 5.2 the identified limitations and constraints related to the currently available FPGA ILAs are detailed with examples. Next, in Section 5.3 the identified solutions for solving the various issues are described. In Section 5.4 a generalized implementation of the proposed FPGA ILA on the FPGA evaluation board used for ROC testing is detailed. It uses one of the available 10 Gbps Ethernet [32] interfaces. The advantages and disadvantages of this solution are explained. Some preliminary experimental results that perfectly match the theoretical expectation are presented in Section 5.5. The research presented in this chapter was disseminated in [171].

5.1 Logic Analyzers (LAs)

The LA is an electronic instrument used to digitize, sample, store, display, decode, debug and analyze windows of multiple concurrent digital signals of interest from real-world digital circuits and systems. In contrast to oscilloscopes, LAs are dedicated to exclusively probe digital signals on many more channels. Even if real-world digital signals are in fact analog, the probing LA channels transform them into digital representations. Oscilloscopes are used to analyze characteristics of signals considered digital (e.g. rise and fall times, fill factor, jitter and skew, etc.). LAs translate the same signals into logical 1's and 0's to determine the data and protocol correctness. LAs contain three main components: (i) at least one probing system; (ii) a processing system and (iii) a user interface. The block diagram of a generic LA with n probing systems is depicted in Figure 5.1.

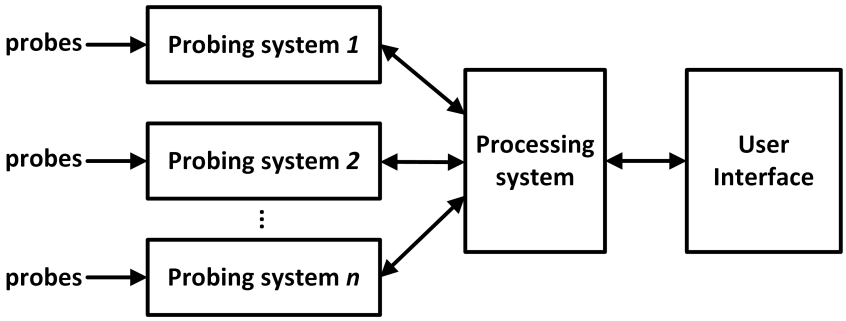


Figure 5.1: The block diagram of a generic Logic Analyzer.

The probing system samples and digitizes the selected signals, affecting the DUT as little as possible (e.g. having high-input impedance). It uses low capacitance probing channels that are connected to the wires associated with the selected design signals. Each LA probing channel can sample a single-ended or differential 1-bit wide signal. In both cases, a configurable comparator circuit translates the voltage level (or the difference between the voltage levels) of the probed signals into an internal signal with the appropriate voltage levels and slew rate. This signal is considered digital, representing a logical 0 if the corresponding probed signal is lower than the established threshold (single-ended) or the negative line voltage is larger than the positive one (for differential signaling) or a logical 1 otherwise. Hysteresis may be also implemented to remove glitches on the resulting internal digital signal caused by a noisy or low slew rate probed signal. Multiple probing channels can be logically grouped (e.g. in the case of parallel data buses). The formed groups are called probes. The resulted binary signals from the probes are sampled by appropriate clock signals which should obey the condition within the frame of the Nyquist–Shannon sampling theorem [158] [180]. The only issue is that a digital signal contains frequencies up to infinity. Most of the time the sampling clock signal has the same frequency as the clock signal that generated the probed signal.

The sampled binary data are buffered by the probing system into memories acting as queues. These memories also allow the safe crossing of the data into the clock domain of the processing system assuming that it is different from the sampling clock domain. For each sampling clock domain, a separate probing system is used. An FSM within each probing system dictates when the data should be discarded or buffered based upon the trigger commands and the conditions imposed by the user through the user interface.

The processing system aggregates the data from all the probing systems, unites the signals into user-defined groups and can detect user-defined patterns, check compliance with standards and protocols and store windows of samples for later inspection. The LA user interface is the component through which the user interacts with the tool. The processing and probing systems are configured through the user interface. The user interface displays the resulted data as timing diagrams with overlays and messages about the decoding and compliance checks.

There are three types of LAs: (i) computer-based; (ii) standalone and (iii) integrated into another electronic instrument or device. The computer-based LAs rely on a host computer to act as the processing system and user interface. In this case, the discrete probing system connects to the host computer through a bidirectional interface. In this way, it receives the user configuration and relays the resulted

sampled data to the host. A standalone LA has a dedicated processing system and user interface (basically a dedicated computer). An integrated LA (ILA) is an accessory that enhances the functionality of an electronic instrument or device, e.g. a Digital Storage Oscilloscope (DSO) that incorporates an LA becomes a Mixed Signal Oscilloscope (MSO). MSOs can therefore display, be triggered on and correlate analog signals, captured by the oscilloscope channels and digital signals, captured by the ILA's channels. Since modern MSOs are computers with specialized hardware and software resources, they can also be considered computer-based. ILAs are also used for debugging, analysis and verification of digital designs implemented in Programmable Logic Devices (PLDs), in particular in FPGAs.

An FPGA ILA is partially implemented into the FPGA's configurable logic. It acquires sample windows of selected signals from the implemented design and transfers them to a host computer where they can be interpreted, checked, stored and visualized as timing diagrams. The entire probing system and part of the processing system are implemented into the FPGA resources unoccupied by the user design. A dedicated application on the host computer acts as the main processing system and the user interface. The acquisition can be triggered manually or automatically when certain user-defined conditions are met, e.g. when an error flag is raised. In the case of Xilinx FPGAs, the user labels the nets to be probed within the RTL HDL code. All these sources are automatically labeled as *do not touch* for the synthesis and implementation tools. The design must be successfully synthesized. Through its GUI or by TCL commands the Xilinx Vivado tool is instructed which of the marked sources of signals are to be sampled, using which clock signals, what is the desired buffer depth and other configuration options. The corresponding ILA is generated as a list of constraints for implementation. The design is then implemented and the bitstream file is generated. Within the Xilinx Hardware Manager tool, when the FPGA device is programmed with this bitstream through the USB-JTAG interface, a dedicated GUI is provided for the user to interact with the ILA. The ILA uses the same USB-JTAG interface for communication with the host computer.

[146] constitutes a good description of the Xilinx FPGA ILA. It defines the LA and includes the idea of an LA incorporated into an IC. It states that an ILA generates a large amount of data that is either stored into an associated embedded memory or is transmitted to an external system requiring high bandwidth. The memory size and/or the available bandwidth constrain the number of probed signals and the probing time. The patent includes drawings and examples of implementations meant to illustrate the claimed, more general novel ideas and not strictly claim the particular implementations.

5.2 Limitations of current FPGA ILAs

The currently available FPGA ILAs can sample only relatively small windows of a limited number of signals from the implemented design, which is not obvious by inspecting their specifications. For instance, one Xilinx ILA v6.2 LogiCORE IP [206] can have up to 1,024 (2^{10}) probes, each from 1 up to 4,096 (2^{12}) bits wide. This is equivalent to a maximum of 2^{22} probing channels or 1-bit wide probes. The Xilinx ILA buffer depth can be set to values in the form of 2^N, with $N \in \{10, 11, ..., 17\}$. The maximum theoretical buffer size for the Xilinx ILA is thus 2^{39} bits = 512 Gb, which seems plenty. But even if the probed signals are paced by a relatively low 100

MHz clock signal, a narrow acquisition window of at most 1.31072 ms is obtained. Similarly, one Altera SignalTap II Logic Analyzer [45] instance can acquire up to 128K (2^{17}) samples from 2048 (2^{11}) channels.

The argument defending this issue is that not all the design signals must be observed at the same time for large time intervals while debugging. It is not practical to monitor large waveform windows, both in number of signals and time interval. One can focus on a specific area of the design and connect the ILA probes to the corresponding signals. However, a re-implementation is necessary if different or new signals must be acquired. As the complexity of digital designs and the clock frequencies increase, the re-implementation and run-times also increase considerably.

In the Xilinx ILA case, the sampled data is buffered into dual-port dual-clock domain BRAMs. The FPGA device most probably will not have such an amount of unused memory available. As a reference example, the Kintex Ultrascale FPGA device used for ROC testing contains 21.1 Mb of BRAM [214] [209]. Even considering that all this memory is available to the ILA and it is not used at all by the implemented user design, it is still four orders of magnitude smaller than the maximum ILA buffer size. Thus, in practice, the number of probed signals and the sampling depth (window size) are merely constrained by the device.

Still, other factors limit the acquisition window depth. For instance, a design occupies 10 Mb of BRAM on the above-mentioned FPGA device and the user only wants to observe the data line of a 64-bit wide AMBA 4 AXI4-Stream interface (i.e. a 64-bit wide signal) [51]. In this case, the sampling depth will be constrained by the Xilinx ILA design and not by available BRAM: the ILA will use a maximum of $2^6 \times 2^{17} = 2^{23}$ bits = 8.39 Mb out of the available 11.1 Mb of BRAM. Even if the depth was not constrained to a maximum of 2^{17}, the ILA design would not be able to take advantage of the entire available BRAM because the depth is constrained to be strictly a power of two.

These memories function as FIFOs whose depths can be divided into a power of two number of windows of equal size (the windows will thus also have power of two depths). For each active trigger signal, a window is filled with the corresponding samples. For the Xilinx ILA, only when a FIFO is full, its content is transferred to the host PC through the JTAG programming and debugging interface. The JTAG speed is board and device-dependent. Furthermore, often the JTAG interface is translated to Ethernet or USB, thus the maximum speed is determined by the slowest interface. While the FIFO content is transferred, new triggers are ignored. This time in which the triggers are ignored is referred to as ILA dead time and it is inversely proportional to the transfer speed to the host computer.

Even if large acquisition intervals are not practical, the user may want to record relatively small windows of samples in response to an event that has increased rate (e.g. the FF and/or SRAM SEU detection in the ROC irradiation test setup from the previous chapter). The already limited buffering space is divided into equal windows. The ILA waits for all these windows to fill and only then transfers the entire buffered data. During the transfer, the new triggers are ignored. A similar situation happens in the Altera SignalTap II LA case. It may be useful to count the triggers even if they are missed, e.g. for determining their rate even without complete corresponding sample data. By extension, the exact moment of trigger occurrence may be provided. These features would have been very useful during the ROC irradiation tests, e.g. for a more precise error count and rate determination.

Thus, the identified issues of the available FPGA ILAs are the following:

1. The sampling depth (i.e. the size of the acquisition window) is constrained by the size of the available unused BRAM on the FPGA device.
2. The sampling depth (i.e. the size of the acquisition window) is constrained to be strictly a power of two.
3. The available BRAM cannot be fully used.
4. Only BRAM is used for buffering and cannot be combined with other sequential resources available on the FPGA device like dedicated FIFOs, distributed memory (from LUTs) and FFs/registers.
5. The queuing algorithm and the relatively slow speed of the interface to the host computer cause increased and in some conditions unnecessary dead-time during the transfer to the host computer which means that triggers are missed and signals of interest are not acquired.
6. Re-implementation of the design is necessary when different signals must be probed which means lost time proportional with the complexity of the design.
7. There is no data about the moments of occurrence or the counts of triggers.

5.3 Proposed solutions

The following solutions that solve or mitigate the previously mentioned issues of current FPGA ILAs are proposed:

1. Use a higher speed interface (compared to the currently used ones like JTAG, USB, etc.) for transmission to the host computer of the ILA-acquired data. Examples of such interfaces are: 5, 10, 40, 100 Gbps Ethernet [32], PCIe [147], Thunderbolt, newer USB versions, etc. The FPGA evaluation board used for the ROC testing and validation contains 8 GTHs (Gigabit Transceivers grade H[1]) dedicated for a PCIe Gen3 ×8 interface and two other GTHs interfaced with Small Form-factor Pluggable (SFP+) network interface modules that can be employed as two separate 10 Gbps Ethernet interfaces [209].
2. Design and implement a new configurable ILA scheduling algorithm that transmits the data while being sampled instead of waiting for the entire buffer to fill and only then flushing its content to the host computer. Furthermore, the proposed algorithm fairly divides and allocates the available transmission bandwidth between the ILA and the user design. Thus, the ILA does not completely occupy the transmission bandwidth of the interface and does not restrict the user design of the usage of such an interface. If the interface is full-duplex and considering the idea of the FPGA-based VE and DUV presented in Chapter 3, Section 3.1, the receive and transmit bandwidths could be shared between the ILA, DUV and VE. If the total throughput of the sampled signals plus the overhead of the Data Link (DL) and Physical (PHY) Open Systems Interconnection (OSI) layers of the interface and the custom on-top protocol for ILA data identification do not saturate the allocated transmission bandwidth and considering that the host computer can store the received data fast enough into its RAM (e.g. through DMA) and not throttle the interface, then the size of the acquisition window is limited by the size of the allocated RAM within the host computer. Otherwise, the size of the acquisition window will

[1] supporting up to 16.375 Gbps [210] [68].

be limited by the available BRAM within the FPGA device. Consequently, the resulted maximum acquisition window size is still larger than in the current state of the art while using FIFOs of the same size because the proposed solution is to send data as soon as it is available and not wait for the FIFOs to fill and then start transmitting. The new ILA scheduling algorithm will thus also decrease or even remove completely the trigger dead-time.

3. Remove the constraint that the depth of the buffering FIFOs must be strictly a power of two. This means that the FPGA ILA is more efficient because it can take advantage of all the available BRAM regardless of its size (assuming that the FPGA ILA design is not constrained by other resources like LUTs, routing, failing timing constraints, etc.). Implicitly the size of the maximum acquisition window will increase.
4. Extend the queuing depth and implicitly the size of the signals acquisition window by combining the available FIFOs and memories of the FPGA: built-in FIFOs, BRAM, distributed RAM (from unused LUTs), FFs and registers.
5. Remove the need for re-implementation when different design signals must be sampled by using the partial reconfiguration feature of the FPGA.
6. Add a timestamp and trigger counter value in each window of sampled data.

The functional features of the current FPGA ILAs are maintained or improved by adding more flexibility. Some examples are:

1. the ability to get data before the trigger happens;
2. the ability to change the trigger condition without re-implementation;
3. the ability to manually trigger the acquisition;
4. the ability to add user-defined complex trigger conditions;
5. the ability to divide the buffer space into windows of equal size which are filled one by one with sampled data every time the trigger fires;
6. the ability to indicate when the data should be sampled from the user design (e.g. through valid or strobe signals).

5.4 Implementation

The block diagram of the proposed FPGA ILA is depicted in Figure 5.2, considering that the implemented user design logic has n different clock domains. Each probing system corresponds to one of the clock domains and contains embedded FIFOs for sample data buffering and logic that determines when and what data should be enqueued based upon the associated configurable trigger logic and the user configuration. The FPGA probing system logic is paced by the same user clock signal that drives the probed logic, to minimize the number of clock domains and the need for implementation of additional Clock Domain Crossing (CDC) logic. The timing constraints for the probing system are the same as for the probed part of the user design. Nevertheless, one must not assume any relationships between the user clock domains and the clock domain of the ILA processing system which is defined by the used high-speed interface. Therefore, CDC must be implemented between each FPGA ILA probing system and the FPGA ILA processing system. This is achieved by employing asynchronous buffering FIFOs. The multiplexer-demultiplexer circuit controlled by the processing system is similar to the ROC's crossbar described in Chapter 2, Section 2.3. The part of the ILA processing system implemented on the FPGA serves the probing systems that contain sampled data, implementing a

data segmentation, multiplexing and acknowledgment protocol (as the Transport OSI layer) on top of the DL OSI layer of the used interface. Since a point-to-point network topology is targeted, there is no need for a Network OSI layer. The DL and PHY layers can be fully or partially integrated within the FPGA logic. A dedicated application runs on the host computer and acts as the main ILA processing system and its user interface.

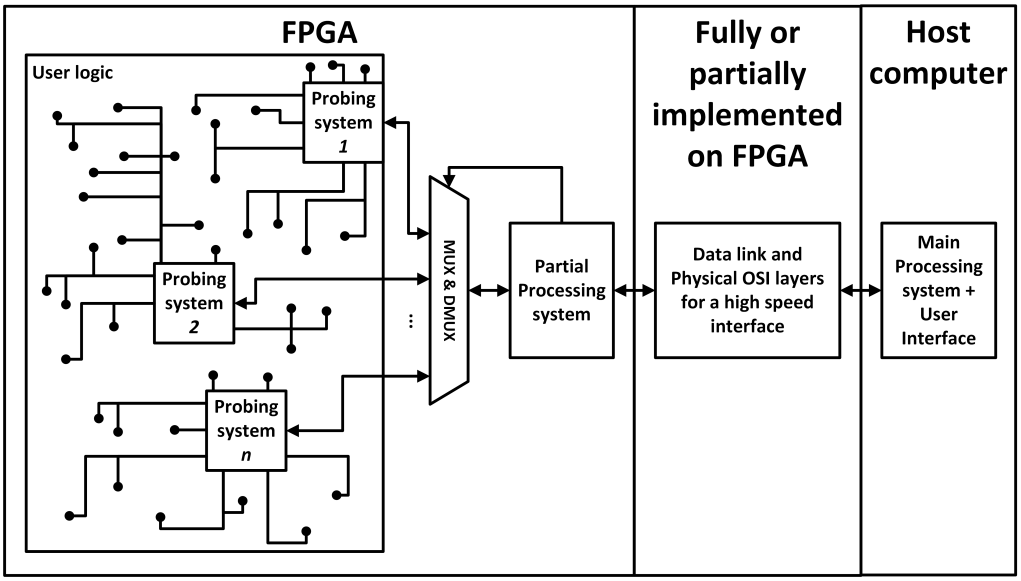


Figure 5.2: The block diagram of the proposed FPGA ILA. From [171] updated.

In Figure 5.3 the proposed top-level architecture of an FPGA ILA which uses an Institute of Electrical and Electronics Engineers (IEEE) 802.3 10 Gbps Ethernet high-speed interface is depicted. The user logic (DUV and VE) has n different clock domains and for each one an FPGA ILA probing system is deployed. The Ethernet interface is shared in both directions with the DUV and the VE. Three variants for the implementation of the Media Access Controller (MAC)[2] and PHY OSI components are depicted: (i) both implemented inside the FPGA; (ii) the MAC sub-layer implemented in the FPGA logic and communicating with a PHY IC from the FPGA hosting PCB and (iii) both implemented outside the FPGA as ICs on the accommodating PCB. The clock signal of the FPGA ILA processing system is shown as having the two possible frequencies of 156.25 and 312.5 MHz because these are the frequencies associated with most standard interfaces of communication between 10 Gbps Ethernet MAC and/or PHY ICs, e.g. AXI4 Stream [51], 10 Gigabit Media-Independent Interface (XGMII) or 10 Gigabit Attachment Unit Interface (XAUI) [32]. The proposed scheduling algorithm for the usage of the 10 Gbps Ethernet interface is Weighted Round Robin (WRR) implemented on a per-quanta basis. The size of the quanta is configurable. The weights allocated to the clients are determined by the throughput ratio of the FPGA ILA probing systems and the DUV and VE logic. The FPGA ILA's custom on-top of Ethernet protocol proposes an Ethernet payload format that allows the multiplexed segmented transmission of sampled data, data from VE, data from DUV, status notifications and the reception of configuration commands and VE and DUV data. The source FPGA ILA probing

[2]the component of the DL layer that is interfaced with the physical layer being responsible for frame synchronization and determining who is allowed to access the media at any one time.

system of the sample data must be easily identifiable. The FPGA ILA application which runs on the host computer in the proposed implementation from Figure 5.3 must also decode this custom format and implement the custom protocol. It must also pass in both directions the corresponding data between the VE application and the VE and DUV.

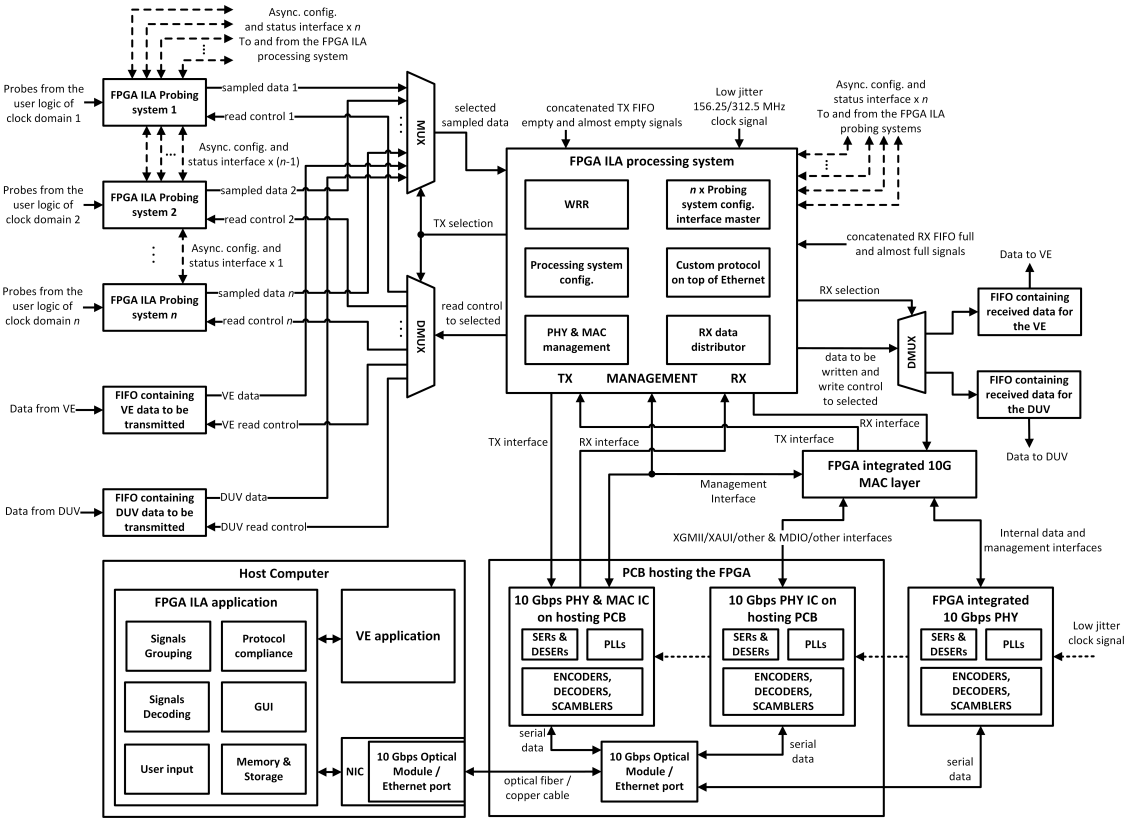


Figure 5.3: The architecture of the proposed FPGA Logic Analyzer. From [171].

The FPGA ILA probing system consists of logic that detects user-defined trigger conditions and buffers the sampled data into the associated embedded FIFOs. The proposed top-level architecture of the FPGA ILA probing system is presented in Figure 5.4 with emphasis on data flow and CDC (assuming that the user clock domain differs from the clock domain of the FPGA ILA processing system). The probing system configuration and status monitoring can be achieved through a register bank accessible through an asynchronous parallel interface. Examples of such asynchronous interfaces are presented in [155], e.g. the common two and four-phase handshaking channels. The FPGA ILA probing system acts as a slave to the master FPGA ILA processing system. All transactions through this interface are initiated by the FPGA ILA processing system. The four-phase handshaking is more reliable while the two-phase one has higher throughput.

For each probed signal or bus from the user logic that is configured to be potentially used as a trigger, a digital comparator circuit is implemented within the trigger sub-system of the FPGA ILA probing system. The comparison is made with values from the configuration registers. Each comparator circuit has two output signals: (i) one high if the values are equal, low otherwise and (ii) the other one high if the value of the probed bus is greater than the value of the register, low otherwise. The user can form basic automatic trigger conditions by applying AND, NAND, OR,

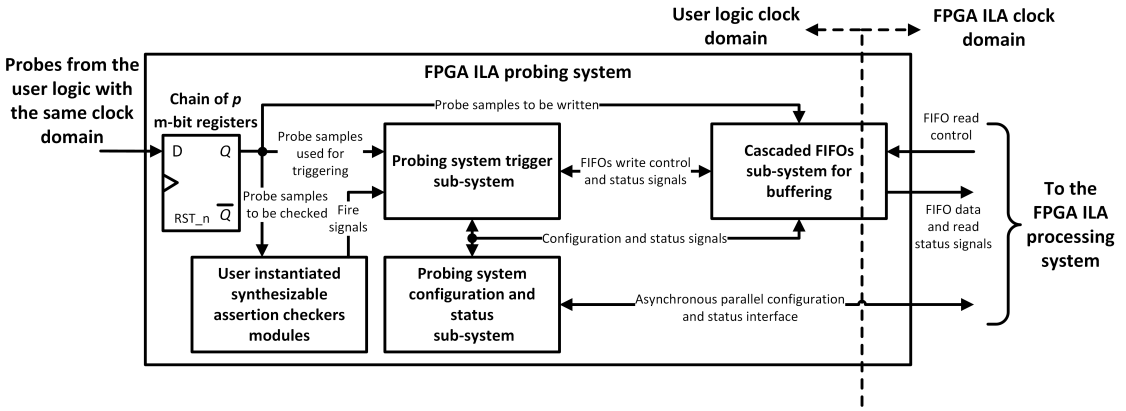


Figure 5.4: The proposed top-level architecture of the FPGA ILA probing system with emphasis on data flow and clock domain crossing.

NOR, XOR or XNOR logic operations on all these signals from all the comparators. Each of the signals can be masked so that it will not contribute to the trigger condition. Custom user-instantiated synthesizable assertion checking modules can be used to trigger the acquisition. Manual triggering is achieved by setting a certain bit of a dedicated probing system configuration register. FFs and/or registers may be added between the probed user signals and the FPGA ILA probing system logic to relax the routing effort and achieve the targeted frequency easier. The constraint is that all the probed signals from a probing system are to be delayed with the same number of clock cycles (i.e. the same number of FFs or registers). In this way, the phase relationship between them is maintained. The probing system depicted in Figure 5.4 samples user design signals from the same clock domain totaling m bits. For each one, a chain of p registers and/or FFs are deployed from the source to the probing system's logic.

FIFOs are used to buffer the sampled data of the FPGA ILA probing system until it is read and transmitted by the FPGA ILA processing system. FWFT FIFOs are recommended for lower latency. This means that valid read data is available at the output whenever *empty* is low and reading the FIFO causes the next word to be available on the following cycle unless the FIFO becomes empty. In a non-FWFT FIFO data is valid on the clock cycle after the assertion of *ren* (read enable) when *empty* is low. The buffering FIFO's logic implemented on the FPGA device technology can be built-in or use Configurable Logic Block (CLB) resources. BRAM, registers or distributed memory (from the unoccupied LUTs) can be used as the buffer memory. The FIFOs have different write and read clock domains (i.e. they are asynchronous) because one cannot assume that user logic uses the same clock domain as the FPGA ILA processing system. Such a FIFO is usually constrained to have only a power of two depth because it uses Gray code pointers to address dual port, dual clock domain memory/memories used for data buffering and CDC, as shown in [90] and [92]. The Gray coding is used to assure that when a pointer increments only one bit will change. This allows a safer crossing of the pointer in the complementary clock domain for comparing and generation of the *empty*, *almost empty*, *full* and *almost full* status signals, as detailed in [90]. An adaptation of this design was used for the VMM Capture and SROC Packet FIFOs within the ROC. The asynchronous FIFO design presented in [92] asynchronously compares

(i.e using only combinational logic) the Gray read and write pointers and then the 1-bit wide resulted signal is crossed into the two clock domains for determining the *empty* and *full* states using two double-FF synchronizers [118]. In this case, the Gray values assure that the glitches of this asynchronous signal are minimized. In the two asynchronous FIFO designs, the Gray coding property must remain true when the pointer increments past the final address. This is always true if the FIFO depth is a power of two. To add more flexibility, a FIFO synchronous to the write domain can precede the asynchronous FIFO. Thus, the total FIFO depth is the sum of the depths of the two FIFOs. The synchronous FIFO is not constrained to use Gray code pointers or to have a power of two depth. However, most memories have a power of two depth. In case the desired total depth is not reached, more synchronous FIFOs can be cascaded with the same method or multiple memories can be used to extend the address space. The resulting total buffer depth is thus not constrained to be only a power of two. Another option is to implement Gray code counters covering non-power-of-two intervals as the write and read pointers of a single asynchronous FIFO. [170] presents such a counter. For each cascaded FIFO sub-system or single asynchronous FIFO, the depth is determined considering the available bandwidth and memory, the WRR quanta size and the allocated weight, the total number of clients for the WRR and the maximum size of the packets that the high-speed interface allows.

In case the user configures the FPGA ILA probing system trigger to be placed at the n^{th} position within the acquisition window (i.e. to have n signal samples before the trigger happens), then the synchronous FIFO is used for buffering the last n samples. When the trigger condition happens, the read samples from the synchronous FIFO are written into the asynchronous FIFO as long as it is not full. The total *full* condition happens when both FIFOs are full. The FPGA ILA processing system knows that data is available from the FPGA ILA probing system when the *empty* status signal of the asynchronous FIFO is low. The additional *almost full* and *almost empty* signals may be useful for the scheduling algorithm of the FPGA ILA processing system (e.g. giving priority to the probing system close to being full). If new trigger conditions are ignored because there is no free memory to buffer the associated sampled data a flag is raised in the configuration and status registers of the FPGA ILA probing system.

As mentioned in Section 5.2, it may be useful to count the triggers and associate a timestamp to each one, even if they were ignored due to insufficient free memory. The trigger timestamp can consist of the value of a clock cycle counter at the moment when the trigger happened. The clock cycle counter starts as soon as the probing system is enabled. The trigger counter could be cleared through an explicit command or self-cleared when the probing system transitions from the disabled to enabled states. It increments in each clock cycle in which the trigger signal is high. The probing system could report the count and timestamp values to the ILA processing system through the configuration and status interface. The interface is designed for relatively low throughput and will not cope with a high trigger rate. Another option is to insert the values between the data buffered in the cascaded FIFO system. In case the trigger is asserted, the cascaded FIFO sub-system is continuously written with sample data. One cannot write two addresses into the cascaded FIFO at the same time. One solution is to store the sample data into a register while pushing the trigger information into the FIFO. In the next clock cycle the data from the

register is pushed and the register is written with the newly sampled data. Until the acquisition window ends, no more trigger information will be written. Another option is to implement an additional asynchronous FIFO which will buffer and pass the trigger information (alongside a bit flag signaling if the trigger was ignored or not) to the ILA processing system. The FPGA ILA processing system will first pop the triggers from this FIFO until it finds one that was not ignored, meaning that the cascaded FIFO system contains its associated data at the output. Then it will pop this sampled data until the end of the acquisition window. This process repeats for the next triggers. This option requires more resources.

In Figure 5.5 the proposed Ethernet payload format for the transmission of ILA data is depicted. The first byte distinguishes between user data (i.e. VE and DUV), ILA status and sampled data. The sample data is segmented in chunks of size q (in bytes), called quanta. In each Ethernet frame, for each probing system, there are as many quanta as its weight dictates. The sum of all weights is W resulting in an Ethernet frame payload that contains W quanta of sample data, provided by $n \leqslant W$ probing systems. Each ILA probing system is uniquely identified by a two-byte field. After each such identifier, the next two bytes indicate the possible start position of a new acquisition window in the quanta. Next, q bytes of data follow, where q is configurable. A new acquisition window always starts with the information of the trigger that prompted the acquisition, consisting of a 4-bytes wide timestamp, 2-bytes wide trigger count and 2-bytes position of the trigger in the window (see probing systems #1 and #W). One quantum can contain data from one or two acquisition windows. The latter is the case for probing system #1 in Figure 5.5: in the $q = p+r+8$ bytes of data, p bytes are from the first acquisition window and r from the second one.

Considering the IEEE 802.3 Ethernet packet structure [32] and the minimum Ethernet payload size of 46 bytes, the quanta q can take values in the range $[\frac{45-W\cdot4}{W}, \frac{s-1-W\cdot4}{W}]$, where s is the maximum payload size in bytes (e.g. $s = 1500$ for standard frames). The effective sample data transfer efficiency for a probing system i with weight w_i, $\sum_{i=1}^{n} w_i = W$, is described by $\eta_i = \frac{w_i \cdot q}{39+W\cdot(4+q)}$, taking into account the minimum Inter-Packet Gap (IPG) of 12 bytes and the other Ethernet packet fields.

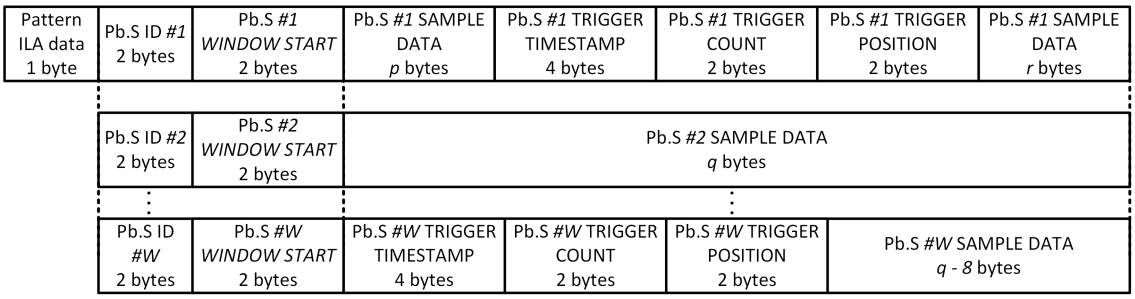


Figure 5.5: Proposed Ethernet payload format for transmission of ILA sampled data to the host computer [171].

The advantages of the proposed solutions are the following:

1. larger sample windows than current FPGA ILAs;
2. reduced or even zero dead-time after a trigger fires;
3. no re-implementation required when different design signals must be sampled by employing partial FPGA reconfiguration;

4. close to real-time FPGA-based DUV verification;
5. long DUV operation time intervals reached faster than in simulations;
6. increased flexibility by using built-in dedicated FIFOs, distributed memory, registers, FFs and BRAMs rather than only BRAM for buffering;
7. more efficient use of the available unoccupied FPGA resources (e.g. FIFO depths not constrained to be powers of two);
8. configurable time-multiplexing of the high-speed interface between the user design and the FPGA ILA;
9. added flexibility by using the receive link of the full-duplex high-speed interface for other communications with the host computer, e.g. for driving internal FPGA signals in real-time or receiving data to be processed by the DUV.

The disadvantages of the proposed solutions are the following:

1. higher cost as it requires an FPGA device and supporting PCB with the resources capable of concurrently accommodating the user design, the ILA logic and the high-speed interface:
   (a) low jitter clock sources for pacing the high rate transfers through the interface to and from the host computer (e.g. the 10 Gbps Ethernet-specific XGMII requires either a 156.25 or 312.5 MHz clock signal depending on its data width - 64 or 32-bit respectively);
   (b) FPGA logic or an external IC (on the supporting PCB) implementing the DL layer of the OSI model for the chosen interface;
   (c) FPGA logic or an external IC (on the supporting PCB) implementing the PHY layer of the OSI model for the chosen interface;
   (d) high-speed transceivers, encoders, decoders, scramblers and PLLs inside the FPGA when internally implementing the DL and/or PHY layers or when interfacing the FPGA to the ICs that implement these OSI layers (e.g. for XAUI between the internal DL and the external PHY IC);
   (e) a relatively large amount of the available IOBs of the FPGA device (e.g. when using XGMII).
2. higher cost as it requires a host computer capable of accommodating the high-speed interface and handling the transferred data, e.g. if the interface is 10 Gbps Ethernet the host computer requires:
   (a) a compatible 10 Gbps NIC with DMA to fast-enough RAM;
   (b) sufficient free RAM for the desired sample window size and number of probed signals;
   (c) sufficient processing power.
3. increased complexity and integration time.
4. higher usage on the FPGA resources: IOBs, FIFOs, BRAMs, routing, interconnects, LUTs and FFs restricting the implementation of the user design and its maximum operating frequencies. 3D stacked and/or bigger (but more expensive) FPGAs can reduce this impact.
5. higher power consumption caused by the increased usage of resources and the higher operating frequencies.
6. tighter design constraints for the PCB hosting the FPGA and the synthesis and implementation of the ILA logic on the FPGA due to the required higher rates and quality of signals.

In [146], several rationales and ideas also presented in this chapter are mentioned:

1. the sampling rate is usually determined by the clock signal that paces the

probed signals;
2. the sampled data is buffered within an embedded memory (e.g. a FIFO) or transmitted to an external system;
3. more resources are required to probe more signals for larger intervals;
4. the ILA constrains the observed system;
5. conventional ILAs use the relatively slow JTAG interface;
6. an ILA that buffers and transmits the sampled data is more complex;
7. in particular cases ILAs are capable of continuously transmitting the sampled data to the external system as a stream, in real-time or close to real-time, while not being limited by the available embedded memory;
8. use timestamps;
9. use user-defined trigger conditions;
10. transmission of data while it is produced;
11. use AMBA AXI4 stream interfaces, cross-bars, network on chips, gigabit transceivers, USB and Ethernet standards; etc.
12. use external IO devices to facilitate the transmission;
13. generation of waveform visualizations within the external interfaced system.

5.5 Experimental results

The theoretical efficiency, as defined in Section 5.4, is plotted in Figure 5.6 as a function of the chosen quanta size q considering a scenario with four probing systems with different weights and no user or status data being transferred while the acquisition takes place. For a steady-state system, the equation $\eta_i \cdot v > \left\lceil \frac{b_i}{8} \right\rceil \cdot 8 \cdot f_i$ must be true for all the probing systems, where v is the speed of the interface to the host in bps and b_i and f_i are the number of 1-bit signals from the user logic being sampled by the probing system i and their frequency, respectively.

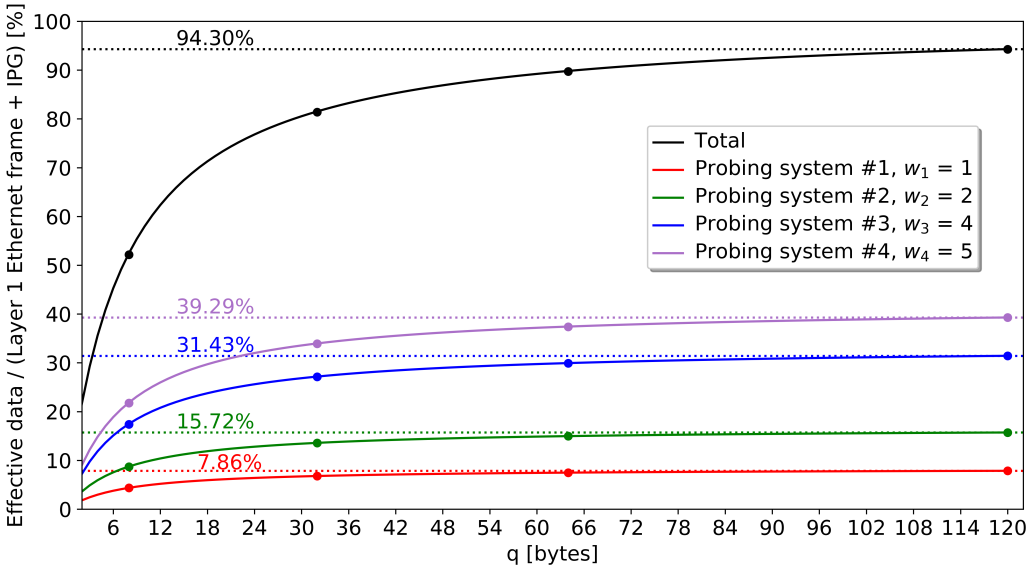


Figure 5.6: Theoretical plots and measurement points of the effective sample data throughput as a function of quanta size q.

The back-to-back transmission of Ethernet packets with the proposed ILA payload format from Figure 5.5 was implemented. Data transfer tests using the scenario

depicted in Figure 5.6 were performed for quanta sizes of 8, 32, 64 and 120 bytes, respectively. The points on the graphs represent measurements of the effective sample data transfer efficiency. These measurements confirm the theoretical hypothesis presented in Section 5.4.

5.6 Conclusions

A time-multiplexed ILA for FPGA that uses a higher-speed interface and a different scheduling algorithm is proposed. An example using a 10 Gbps Ethernet interface is presented. The main advantage of the proposed design is the ability to sample larger windows of signals than current JTAG-to-USB-based FPGA ILAs. The dead time after a trigger fires is reduced and in some conditions even eliminated. The design adds flexibility by using and combining built-in FIFOs, distributed memory, registers and BRAMs rather than only BRAM for data buffering. It implies a more efficient use of the available FPGA resources (e.g. FIFO depths not constrained to be powers of two). The design is suitable both for close to real-time hardware-in-the-loop verification and reaching long DUV/DUT operational intervals faster than in simulations. By using partial FPGA reconfiguration, no re-implementation is required when other design signals must be sampled. Also, the designed ILA can transmit the data while it is being sampled instead of waiting for the buffer to fill, stopping the sampling and triggering and then flushing the buffer content to the host computer. In addition, the proposed ILA can share the used high-speed interface with the user design. Because most interfaces are full-duplex (e.g. the 10 Gbps Ethernet interface), the FPGA's receive link can be used for other communications with the host computer, e.g. for driving internal FPGA signals in close to real-time or receiving data to be processed by the DUV.

The main disadvantage is that the proposed design requires an FPGA device with resources capable of concurrently accommodating the user design, the ILA logic and the high-speed interface. Other resources are required: low-jitter clock sources for pacing the high rate transfers and ICs implementing DL and/or PHY OSI layers. If the FPGA internal logic also implements the DL and PHY layers of the interface, the device requires high-speed transceivers, encoders, decoders, scramblers and PLLs. A relatively large amount of the available IOBs must be dedicated when using the XGMII in the 10 Gbps Ethernet case. Other disadvantages include tighter constraints upon the user design and higher power consumption caused by the increased usage of resources and the higher operating frequencies. The proposed design has increased complexity and requirements. Last but not least, a host computer capable of accommodating the high-speed interface and handling the transferred data is required.

The proposed FPGA ILA would have been a very useful tool during the ROC's irradiation testing since it enhances the observability of the test setup. The results and analysis from Chapter 4 would have been consequently improved. Unfortunately, Xilinx and other companies directly interested in this subject protect their inventions with patents but are not bound to implement and provide them to the user of their tools and/or devices. Nevertheless, the proposed high-speed time-multiplexed FPGA ILA resulted as an application/consequence of the ROC's mass-testing and irradiation qualification.

Chapter 6

Conclusions

This chapter summarizes the conclusions from all the other chapters while emphasizing the author's contributions and achievements related to the thesis subject. Then the dissemination of results, including an enumeration of publications, is presented, as well as various communications on related topics.

6.1 Final Conclusions

All the subjects touched by this thesis relate to the high luminosity general-purpose ATLAS Experiment from LHC. The LHC, operated by CERN, Geneva, Switzerland, is the largest and highest-energy artificial synchrotron in the world. The experiments at CERN (including ATLAS) are used for fundamental research in particle physics, mainly to confirm, correct and/or extend current models and theories like the SM and SUSY or even formulate new ones that better describe the matter of the Universe. To detect and characterize all the products resulted from the particle beam's collisions within ATLAS, an advanced detector system and an associated TDAQ system were designed and successfully used, leading to the confirmation of the theorized Higgs boson in 2012. The LHC follows a sequence of operational runs that alternate with maintenance and upgrade processes for the accelerators, detectors and the TDAQ systems.

The current upgrade process implements the first phase of the HL-LHC project which aims at reaching 3000 fb^{-1} integrated luminosity at 14 TeV energy in 10 years of operation translating to one order of magnitude increase in size for the collected data. To cope with the increased nuclear radiation and the necessity for more precise discrimination and characterization of the collision's effects taking place at higher rates, the ATLAS TDAQ system, the associated software tools and part of the detector system are being upgraded. Some of the most important improved subsystems of the ATLAS detector are the two NSWs, part of the Muon Spectrometer. They use two new detector technologies to more precisely determine the muon tracks originating from the interaction point while also reducing the rate of false triggers and the performance degradation at high energies and luminosities that characterize the previous detectors. The associated NSW TDAQ system interfaces with these new muon detectors and the rest of the ATLAS TDAQ system and is designed to cope with the increased radiation, data throughput, rates and latencies.

Part of the NSW readout path, the ROC is an on-detector radiation-tolerant

ASIC acting as a real-time concentrator, buffer, filter and packet data processor. The presented work represents the author's contribution to its design, verification, implementation, quality assurance and control, support for integration and documentation, including its mathematical and algorithmic descriptions. The author's work was performed while being a research assistant in the "Experimentul ATLAS de la LHC" (i.e. The ATLAS Experiment from LHC) national research project as member of the team from Transilvania University from Brașov.

The performed work covers: digital electronics design, digital circuits simulation, digital circuit verification, programming, data structures and algorithms, data communications, embedded systems, EDA tools, operating systems, scripting, statistics, signal processing, packet processors, network-on-chip, measuring instruments and scientific and technical document preparation. The work implied extensive real-world debugging, equipment use and measurements. In addition, the author attended special courses presented by prestigious scientific research centers[1] [2]. Other covered topics include but are not limited to: modular redundancy, schematic PCB design, digital circuit synthesis, digital circuit constraining, static timing analysis, design for testability, logic equivalence checking for digital circuits, gate-level simulations of digital circuits, ASIC floor-planning, ASIC placement and routing, digital circuit testing, goodness-of-fit assessment, scientific collaboration, research project management and dissemination of scientific results.

The author contributed to the development and verification of the VMM3 L0 event building logic; the development, verification and implementation of the ROC digital part's logic and the development and verification of the configuration logic for the ROC's analog part. Based upon this work, small complications in the protocols and inconsistencies between the VMM3 and the ROC were discovered and solved before the tape-out.

The ROC design was successfully taped out for fabrication on time, on the 15$^{\text{th}}$ of August 2016. Between the sign-off and the arrival of the prototype for design validation, the author contributed to the documentation of the ROC digital design, the construction of its exhaustive functional test checklist, the development of the schematic for the first testing PCB and of the FPGA-based functional test setup for the digital part. A functional test setup was chosen because it can be used for the digital design validation, mass testing and radiation qualification and because the ROC's design does not include scan chains. The first ROC ASIC was powered up in the test setup on the 24$^{\text{th}}$ of March 2017, after it arrived at Brașov through heavy snow. Back-to-back commas being transmitted at the default highest speed were detected and recorded from its output serial lines. When VMM3 packets were injected in the input channels in conjunction with issuing the corresponding TTC commands, output packets were recorded but error flags were raised. The fact that the chip was active was a good sign. Since then, the test environment was massively improved. All the ROC's functional features were covered. The main causes for the initial errors were setup and hold violations inducing meta-stability in the capturing FFs from both the ASIC and the FPGA design. The high-speed serial lines between

[1] "Comprehensive Digital IC Implementation & Sign-Off" held in November 2015 by the Microelectronics Support Centre Science and Technology Facilities Council from Rutherford Appleton Laboratory (RAL), Oxford, Didcot OX11 0QX, United Kingdom.

[2] "Verification for Digital Designs" held between 4 - 6 December 2019 by the Microelectronics Support Centre Science and Technology Facilities Council from RAL.

the ASIC and the FPGA required calibration and the issues were solved.

The ROC digital design was also tested and validated in parallel by a team from University of Michigan, part of the NSW collaboration, using the same design for the accommodating PCBs and the same type of FPGA evaluation boards but a different, separate, in-house FPGA design based upon the chip's specifications. The author provided support to the Michigan team for understanding the ROC's specifications and design. The ROC's digital design was validated by both sides but some bugs that were not design-breaking were discovered in the digital part. The Michigan team also thoroughly analyzed the ROC's analog part. In parallel, at Horia Hulubei National Institute for Research and Development in Physics and Nuclear Engineering (IFIN-HH), Măgurele, Romania, the ROC's analog part was also tested. An uncertain phase after power-up for each output clock signal relative to the reference BC clock signal was discovered [61]. Thus the ROC's ePLLs macros were redesigned at IFIN-HH while the associated configuration logic (except for the reset values for some registers) and the digital part remained the same. The author's contribution to the redesign was limited to running and validating SDF back-annotated gate-level simulations of the ePLL configuration and monitoring logic interfaced with behavioral models for the redesigned ePLLs.

The redesign solved the phase uncertainty for the supplied clock signals and the new chip was named both ROC1A and ROC2 [109]. Its mass fabrication was achieved on an MPW alongside VMM3A (i.e. a small redesign of the VMM3's analog part that digitizes the MM and sTGC detector signals) and other NSW ASICs, as shown in Figure 6.1. Once it was confirmed that the analog redesign solved the phase uncertainty, the AF region was also occupied by the ROC2 design. It was decided that the quality control of the fabricated samples will be carried out using the digital test setup presented in Chapter 3 and a separate automated analog test setup developed at IFIN-HH, Măgurele. This analog test setup uses the same type of FPGA evaluation board as the digital one and motivated the design of the fifth ROC testing PCB version.

In the summer of 2018, the ROC was successfully tested while being hit by ultra-fast controlled neutron beams produced by the TANDEM accelerator from NCSR Demokritos, Athens, Greece. The chip required a basic radiation qualification process since it was part of the on-detector ATLAS electronics. The ROC chip's operation was not directly disturbed by the incident radiation due to the implemented TMR but the data corruption in its SRAMs occurred at a considerable rate and caused data loss. Estimations as accurate as possible were made for the HL-LHC. At the highest neutron energy (i.e. 24 MeV nominal) the rate of induced effects was significantly higher than at lower energies. Insights on the possible causes and the method of producing estimations were provided by Dr. Theodoros Geralis, Director of Research at the Institute of Nuclear and Particle Physics at NCSR Demokritos, Athens, Greece, by Prof. Theodoros Alexopoulos and Prof. Kokkoris Michael from National Technical University of Athens, Greece and by Assoc. Prof. Thomas Schwarz from University of Michigan, USA. The idea of a high-speed FPGA ILA with a different operation than the provided one was born from the lessons learned during these experiments.

At Transilvania University of Brașov, a total of 2677 ROC1A/ROC2 chips were tested using the two functional test FPGA designs (i.e. analog and digital) replicated in up to three test benches. Over half of these samples were tested by the author of

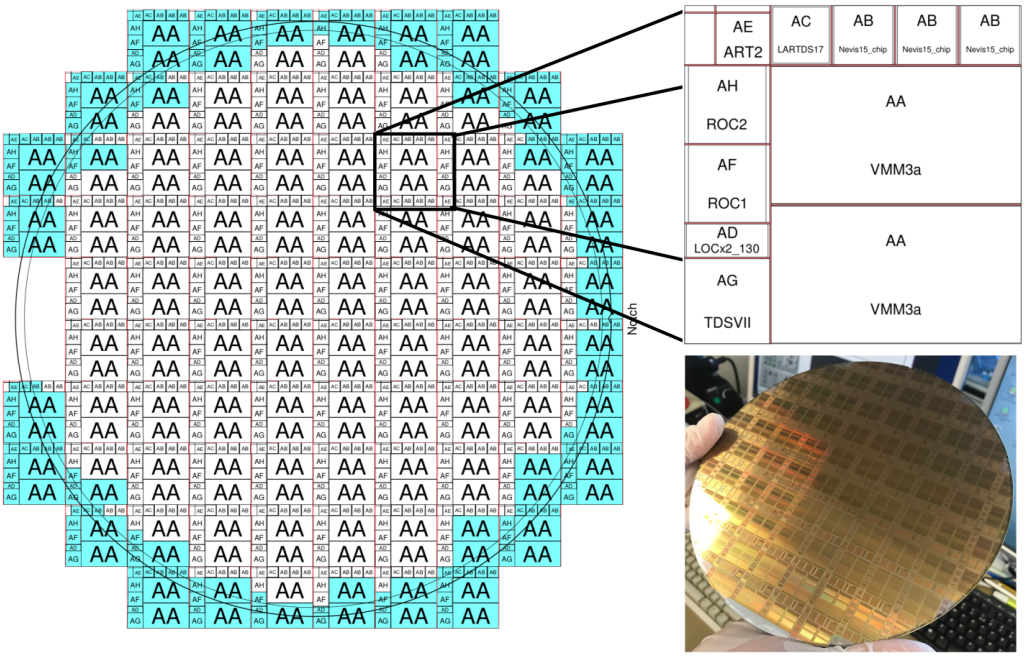


Figure 6.1: MPW containing NSW ASICs including the two versions of ROC [131].

this thesis. As a reference, at least 4875 ROC1A/ROC2 ASICs will function simultaneously within the NSW TDAQ system. The test setup was not fully automated and required periodic human intervention, e.g. for the insertion and extraction from the socket and sample identification. Other test benches were deployed and successfully used at IFIN-HH and INCDTIM[3], Cluj-Napoca, Romania. The initial yields were considered unsatisfactory, as the majority of chips were failing the digital test. The author concentrated his effort on the calibration procedures for the high-speed transmission lines between the ASIC and the FPGA. This also motivated the development of a faster calibration algorithm. The final digital testing yields (86 - 89 % at nominal voltage and 61 - 73 % at sub-nominal supply voltage) were considered satisfactory [109].

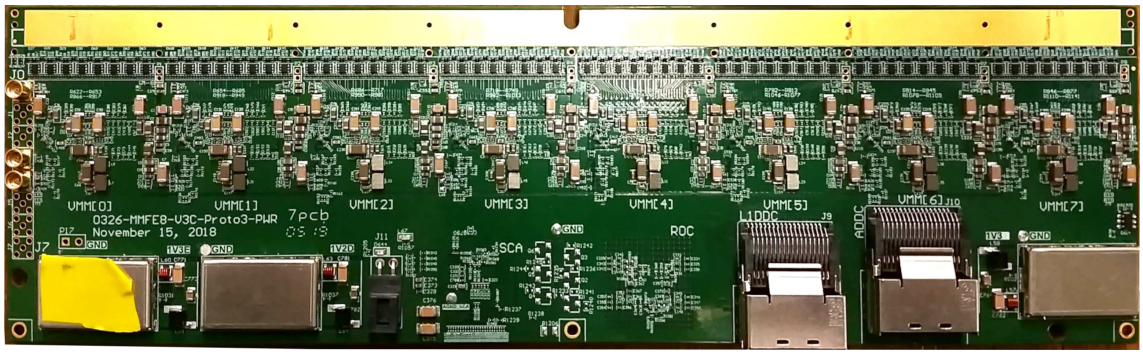
The author periodically reported the progress of the work and the testing results in NSW coordination meetings. As an achievement and recognition for his contribution to the Experiment, he completed the one-year ATLAS author[4] qualification procedure in November 2018. The thesis author actively participated and still is participating in the ROC integration efforts within the NSW collaboration. In 2018, two mobilities at CERN were done for this specific purpose.

The ROC1A/ROC2 design passed all the NSW internal reviews and is included in its readout system as shown in Figures 6.2 and 6.3. Published preliminary integration results are beginning to appear [190].

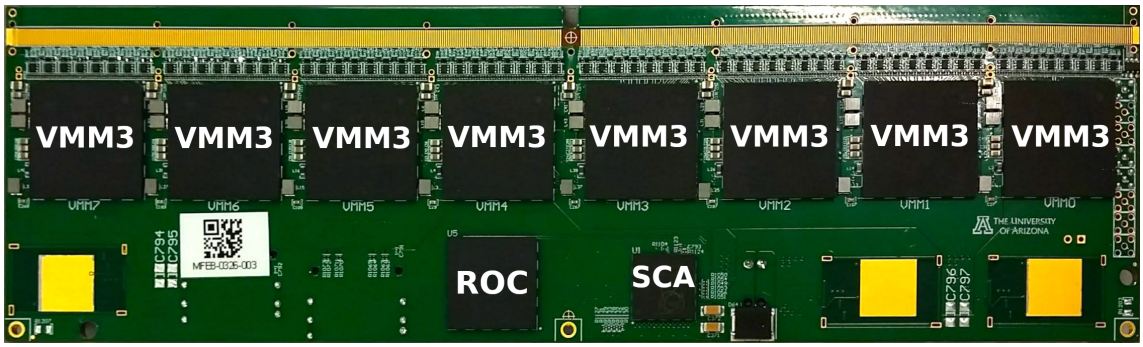
As questions on the ROC's performance in specific scenarios often arise, the author proposed and validated a mathematical model for the maximum data rates that do not induce loss (deduced in Chapter 2, validated in Chapter 3).

[3]Institutul Naţional de Cercetare Dezvoltare pentru Tehnologii Izotopice şi Moleculare

[4]An ATLAS author is a person officially working on ATLAS who has made significant contributions to the experiment [2]. All ATLAS General Publications are signed by all active ATLAS authors (e.g. [80]).



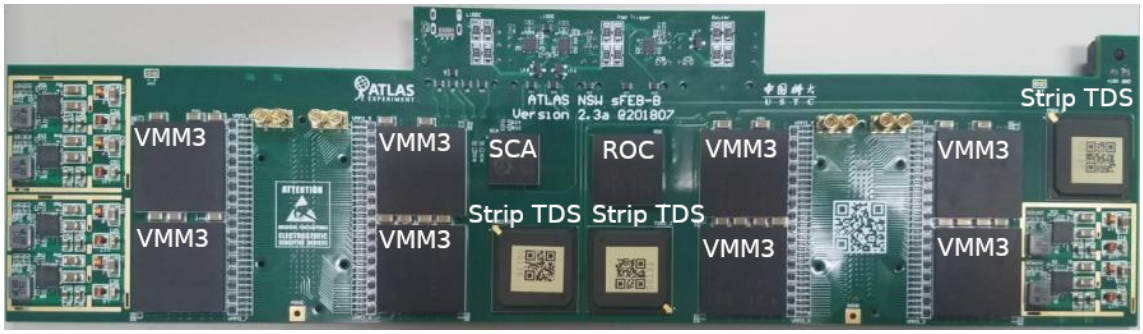
(a) Front.



(b) Back.

Figure 6.2: The MMFE8 containing eight VMM3 chips, one ROC and one SCA (based on original photos by Anne Fortman, Harvard University ATLAS group). The ASICs are highlighted.

(a) pFEB.



(b) sFEB.

Figure 6.3: The NSW front-end boards containing ROC chips on the sTGC path. From [197] with highlighted ASICs.

6.2 Contributions

The author's contributions to the ROC's logic and its quality assurance (i.e. thesis objective 1 from Chapter 1, Section 1.8) are:

- partial elaboration of the architecture and organization (including the description in synthesizable RTL Verilog code) of the following ROC modules:
  - the 8b10b decoder from the VMM Capture (i.e. VMM DEC in Figure 2.3) and the SROC 8b10b encoder (i.e. SROC ENC in Figure 2.7) based upon open-source IP cores, by optimizing and correcting them and adding data validation logic;
  - the VMM Capture, TTC and SROC Packet FIFOs, based upon [90], by adding almost full signals, making them FWFT and compatible with the SRAMs, adjusting their depths and the buffering formats;
  - the SROC Packet Builder FSM and its complementary logic as depicted in Appendices A.1 and A.2, by adjusting the algorithm for all the required functional features and their cohesion and implementing the *bypass* mode, the handling of the special *PATTERN* BCID, the dummy hit words insertion, the logic associated with the *VMM missing* flags, the timeout feature, the parity checking and handling of altered L0 packets, the checksum and parity computation and the logic for the L1 data format generation. The transitions between states were optimized to increase the processing throughput (see Chapter 2, Section 2.4.4).
  - the SROC Streamer FSM, by adjusting its algorithm for all the required protocol functional features and their cohesion and optimizing the transitions between states to maximize the output throughput (see Chapter 2, Section 2.4.3);
  - the TTC Capture module by adding the configurability for the interpretation of the bits from Figure 2.13 and the parallel debug ports (see Appendix A.3);
  - the register bank within the Config module of the digital part by extending and adjusting a template generic register bank to the required functional features, e.g. clear upon reading for certain addresses, status flags and their CDC, counters for SEUs and L0 header parity errors, the separation of the configuration into distinct output signals, etc.;
  - the register bank within the ePLL Config module from the ROC analog part, by adjusting it to the required functional features, similar to the work performed for the digital part's register bank.
- performing the integration and interfacing of the ROC modules into higher-level modules (e.g. VMM Capture, SROC, Config, etc.), following a bottom-up strategy, until the top-level module representing the entire ROC digital part was fully constructed (see Figures 2.1 and 2.2);
- increasing the debugability of the ROC digital part by adding additional IO ports and logic (see Appendix A.3);
- fully elaborating the VEs for the following ROC modules: VMM DEC, VMM Capture FIFO, VMM Capture, SROC Packet FIFO, SROC Packet Builder, SROC ENC, SROC streamer and SROC;
- elaborating the majority of the VEs dedicated to the top-level ROC digital part and Config modules;

- running the verification simulations with the above mentioned VEs and implementing the necessary corrections for the identified bugs;
- elaborating a VE containing the RTL of the L0 packet constructing digital logic from the VMM3 interfaced with the RTL of a VMM Capture channel and running simulations with it;
- proposing minor changes in the ROC specifications that were accepted: the avoidance of the *K.28.7* 8b10b special symbol which requires a more complex synchronization since it forms false *commas* (i.e. *K.28.5*) with other 8b10b symbols and correspondence of the quoted *PATTERN* BCID with the one from the VMM3 logic (as a result of the work mentioned in the previous item);
- applying TMR for all the FFs and registers from all the ROC's digital part modules (see Chapter 4, Section 4.2.1).

The author's contributions to the ROC's implementation into the targeted 130 nm CMOS technology (i.e. thesis objective 2 from Chapter 1, Section 1.8) are:

- constructing the SDC statements related to the preservation of the TMR logic;
- constructing the SDC statements related to timing for the synthesis and physical implementation ASIC design steps;
- performing the synthesis and LEC for the ROC digital part;
- actively assisting to the physical implementation of the ROC digital part: the IO placement and the solving of timing and DRC violations;
- adjusting the top-level VE of the ROC digital part for gate-level simulations with back-annotated propagation delays;
- running all the gate-level simulations with the post-synthesis, post-placement, post-routing and sign-off netlists of the ROC digital part, back-annotated with propagation delays (i.e. from SDF files) in all the available corner cases for the implementation technology;
- running gate-level simulations with the post-synthesis, post-placement, post-routing and sign-off netlists of the ROC analog part, back-annotated with propagation delays (i.e. from SDF files) and using behavioral models for the ePLL macros in all the available corner cases for the 130 nm CMOS implementation technology;
- partially elaborating and verifying the ROC's top-level wrapper Verilog module which aggregates and interfaces the IO and power pads and the analog and digital sub-designs.

The author's contributions to the functional validation of the manufactured design (i.e. thesis objective 3 from Chapter 1, Section 1.8) are:

- partially elaborating the schematics of the first version of the ROC testing PCB, schematic that was the basis for all the other more advanced versions since backward compatibility was maintained (the HPC FMC pin association propagated to all versions);
- elaborating and implementing the majority of the asynchronous FPGA digital functional test setup: the custom C program running on the MicroBlaze microprocessor and all its custom modules except the two I^2C masters;
- validating the ROC digital design in the digital functional test setup;
- improving the functional digital test setup up to a real-world stable state with deterministic and repeatable behavior and response;
- performing the necessary and requested measurements with thermal imagining cameras, DSOs, LAs and FPGA ILAs;

The author's contributions to the ROC chip's performance measurements (i.e. thesis objective 4 from Chapter 1, Section 1.8) are:

- deducing and validating the ROC's mathematical model (see Chapter 2, Section 2.4) in response to the frequent questions from the NSW community about the maximum trigger rates without data loss in different scenarios;
- performing the necessary and requested signal quality measurements (e.g. jitter) with DSOs and LAs;
- performing power consumption and thermal testing.

The author's contributions to the quality control of the mass-fabricated ROC chips (i.e. thesis objective 5 from Chapter 1, Section 1.8) are:

- implementing ten exhaustive tests for mass-testing;
- automating, improving and optimizing the calibrations procedures for the high-speed serial data signals between the FPGA device and the ROC ASIC;
- mass-testing over half (approx. 1500) of the ROC chip samples arrived at Transilvania University of Brașov;
- analyzing the results of the mass-testing at Transilvania University of Brașov and establishing the main causes of failure (see Chapter 3, Section 3.2.4).

The author's contributions to the real-world ROC nuclear radiation qualification (i.e. thesis objective 6 from Chapter 1, Section 1.8) are:

- modifying and adjusting the asynchronous FPGA-based digital functional test setup for continuous ROC testing with its FIFO buffers filled as much as possible without loss of data;
- testing the radiation tolerance of the ROC ASIC while operational in functional mode in an environment with controlled ultrafast neutron irradiation;
- analyzing the results of the radiation tolerance test and computing statistical predictions for the actual operating environment;
- running exhaustive simulations with induced bit-flips within the ROC's SRAMs and analyzing their effects.

The author's contributions to the integration of the ROC ASIC within the NSW TDAQ system (i.e. thesis objective 7 from Chapter 1, Section 1.8) are:

- elaborating (partially), maintaining, correcting and updating the ROC digital logic documentation: IO ports description, internal architecture of the developed modules, register bank map, FSM transition graphs, etc.
- offering continuous ROC-related support to the NSW collaboration.

Regarding the identification and pursuit of new research opportunities during the ROC-related work, the author has the following contributions:

- elaborating an optimized algorithm for the calibration of the clock and data signals within the asynchronous ROC's digital functional test setup, implementing and validating it in real-world;
- identifying the limitations and constraints of current FPGA ILAs from Xilinx/AMD and Altera/Intel and proposing, designing and partially validating the architecture of a new FPGA ILA that mitigates them with a different mode of operation and a 10 Gbps Ethernet interface to the host computer.

The ROC's very specific use means that it is unique/original as a whole even though it is comprised of pre-designed elements (e.g. the used standard-cell libraries, SRAMs, ePLLs, IO pads, etc.). Hence, it represents an important original contribution to a prestigious fundamental research experiment.

6.3 Dissemination of results and training

The scientific publications related to the presented work, corresponding to the thesis objective 9, in the reverse chronological order of publication and with the author's name emphasized are:

1. **Ș. Popa**, M. Ivanovici, R.-M. Coliban, "Time-multiplexed 10Gbps Ethernet-based Integrated Logic Analyzer for FPGAs", *International Symposium on Electronics and Telecommunications, ISETC 2020*, Timișoara, 5-6 November 2020, `https://doi.org/10.1109/ISETC50328.2020.9301115` [171]
2. **Ș. Popa**, S. Mărtoiu, M. Ivanovici, "Study of the ATLAS new small wheel read-out controller ASIC in a neutron irradiation environment", JOURNAL OF INSTRUMENTATION, Volume 15 P10023, October 2020, `https://doi.org/10.1088/1748-0221/15/10/P10023` [174]
3. **Ș. Popa**, S. Mărtoiu, M. Ivanovici, "The quality-control test of the digital logic for the ATLAS new small wheel read-out controller ASIC", JOURNAL OF INSTRUMENTATION, Volume 15 P04023, April 2020, `https://doi.org/10.1088/1748-0221/15/04/P04023` [173]
4. **Ș. Popa**, M. Luchian, M. Ivanovici, "Clock and data signals synchronization for an FPGA-based ASIC testing setup", *14th International Symposium on Signals Circuits and Systems, ISSCS 2019*, Iași, România, July 2019, `https://doi.org/10.1109/ISSCS.2019.8801780` [172]
5. **Ș. Popa**, S. Mărtoiu, M. Luchian, R.-M. Coliban, M. Ivanovici, "The Quality-Assurance Test of the ATLAS New Small Wheel Read-Out Controller ASIC", *Topical Workshop on Electronics for Particle Physics, TWEPP 2018*, Antwerp, Belgium, 17-21 September 2018, `https://doi.org/10.22323/1.343.0081` [175]
6. R.-M. Coliban, **Ș. Popa**, T. Tulbure, D. Nicula, M. Ivanovici, S. Mărtoiu, L. Levinson, J. Vermeulen, "The Read Out Controller for the ATLAS New Small Wheel", JOURNAL OF INSTRUMENTATION, Volume 11 C02069, February 2016, `https://doi.org/10.1088/1748-0221/11/02/C02069` [79]

Papers 2, 3 and 6 are published in an international peer-reviewed scientific journal dedicated to the instrumentation for detector and accelerator science, indexed in the Web of Science database, having an impact factor of 1.454 in 2021. Papers 1 and 4 were published in the proceedings of international scientific conferences organized in Romania, conferences that are also indexed within the Web of Science database. Paper 5 is part of the proceedings of the Topical Workshop on Electronics for Particle Physics (TWEPP) 2018 where the author also participated with a poster (link[5]: `https://indico.cern.ch/event/697988/contributions/3056039/`). This workshop/conference is not indexed within the Web of Science database but in other prestigious databases like Scopus. In 2015, a colleague in the research project participated in that year's edition of TWEPP with a poster about the VMM3 and ROC's role within NSW and their digital architecture (link[5]: `https://indico.cern.ch/event/357738/contributions/848825/`). Even if the author contributed to the poster, his name does not appear on it. However, on the contribution page (i.e. previous link) all the authors are listed.

Thus, three journal papers and three conference papers, all indexed in Web of Science except for item 5, were published.

[5]requires a CERN account.

Paper 6 presents the ROC's context, data formats and design. Papers 2 and 3 also include such descriptions for completeness. Thus, the research presented in Chapter 2 is disseminated through these three papers. Paper 3 presents the ROC digital functional test setup, its design and performance validation and the mass-testing results. Paper 4 describes the improved synchronization method for the clock and data signals within the ROC's digital test setup. Thus, the research presented in Chapter 3 is disseminated through these two papers. The ROC's radiation qualification presented in Chapter 4 is disseminated through paper 2. The research related to the FPGA ILA proposed in Chapter 5 is disseminated through paper 1.

Other forms of dissemination are:

- The results of the author's work were also presented in multiple online meetings of the NSW Electronics group. The majority of them are available in the ROC's dedicated shared directory[6], alongside all the documentation related to the ROC digital logic, the majority of which was also the result of the author's work.
- The author proved and defended the validity of the ROC ASIC design and the corresponding results in front of many evaluating commissions (NSW electronics group, ISAB CERN-RO - Întrunirea Consiliului Științific International Consultativ CERN - "Meeting of the CERN International Scientific Advisory Board" in English, BSc, MSc, reviewers from the conferences and the journal where the papers listed above are published, etc.), integration teams and users of the ROC ASIC.
- The author held presentations and posters at AFCO (Absolvenți în Fața COmpaniilor - "Graduates in Front of Companies" in English)[7] and SCSS (Sesiunea Cercurilor Științifice Studențești - "Scientific Student Circles Session" in English)[8] in 2015 and 2017;
- The author held a presentation about his work in the Young Scientist Forum at the ISAB CERN-RO meeting on the 26$^{\text{th}}$ of October 2017.
- The author also presented his work at the European Researchers' Night editions from 2016, 2017 (both hosted at Andrei Șaguna National College from Brașov) and 2018 (hosted at the Research and Development Institute of Transilvania University of Brașov).
- The author presented his work at the annual ATLAS group meetings from 2017, 2018 and 2019.
- The author presented his work at PatriotFest 2017[9].
- The author participated in the workshops of Prof. Loius François Pau (Copenhagen Business School and Erasmus University – Denmark) on the 26$^{\text{th}}$ and 27$^{\text{th}}$ of March 2019 at Transilvania University of Brașov with the subjects: copyright, intellectual property and research paper publishing.

[6]Link requiring CERN account: `https://espace.cern.ch/ATLAS-NSW-ELX/_layouts/15/DocSetHome.aspx?id=/ATLAS-NSW-ELX/Shared%20Documents/ROC&Source=https%3A%2F%2Fespace%2Ecern%2Ech%2FATLAS%2DNSW%2DELX%2FSitePages%2FHome%2Easpx`

[7]An annual event organized by Transilvania University of Brașov and sponsored by local companies with the aim of initiating the dialogue between students and the local industry.

[8]An annual event organized in all the faculties from Transilvania University of Brașov in which any student can present the results of his/her work within research projects.

[9]National contest for innovation for national security.

Appendix A

Appendices

A.1 SROC packet building pseudo-code

Function definition	**Description**
first_VMM()	Returns the index of the first VMM Capture channel associated with the SROC (i.e. the one with the smallest index).
wait_one_clock_cycle()	Waits for one BC clock cycle. Placeholder for doing nothing.
read_TTC_FIFO()	Returns the first L1 trigger from the TTC_FIFO (i.e. pops one word) as a structure of type l1_trigger_type. TTC_FIFO must not be empty.
write_null_event(*VMM_Id*)	Pops the empty header (i.e. the L0 null event) currently present at the output of *VMM_Id* VMM Capture channel (i.e. on *VMM_FIFO[VMM_Id].rdata*), constructs and pushes an L1 null event into the *SROC_FIFO* using the *ROC_Id* setting and the *trigger.L1ID* value. Returns only after the L1 null event is successfully pushed (i.e. if *SROC_FIFO.full* == 1 it will wait until one word is freed and then pushes the null event).
read_null_event_or_header (*VMM_Id*)	Pops the first data word (an L0 null event - empty L0 header) from the VMM Capture channel with index *VMM_Id* (i.e. through *VMM_FIFO[VMM_Id].rdata*).
write_header(*VMM_Id*)	Transforms the L0 header currently present on the output of the *VMM_Id* VMM Capture channel (i.e. on *VMM_FIFO[VMM_Id].rdata*) into an L1 header that is written into the *SROC_FIFO*. The function returns only after the L1 header is successfully written (i.e. if *SROC_FIFO.full* == 1 it will wait until one address is freed and then pushes the header.
write_trailer (*vmm_missing_bits*)	Constructs the trailer word and pushes it into the *SROC_FIFO*. The function returns only after the trailer word is successfully written (i.e. if *SROC_FIFO.full* == 1 it will wait until one address is freed and then pushes the trailer. Uses the *hit_number* and *TO_bit* global variables and the *vmm_missing_bits* parameter.

write_hit_data(*VMM_Id*, *vmm_missing_bits*)	Pops all the L0 hit words of the current VMM3 packet from the *VMM_Id* VMM Capture channel FIFO and transforms them into L1 hit words that are pushed into the *SROC_FIFO*. The function returns only after all the hit words are successfully written (i.e. if *SROC_FIFO.full* == 1 it will wait until at least one address is freed and then pushes the hit words, one by one). For each popped/pushed hit word, it increments *hit_number* and updates *checksum*. It receives the *vmm_missing_bits* and returns the updated value, with a set bit corresponding to the *VMM_Id* VMM Capture channel if any of the L0 hits have a set T bit.
all_VMMs_timeout()	Returns the index of the first VMM Capture channel associated with the SROC (i.e. the one with the smallest index).
write_header_no_read()	Same as write_header(*VMM_Id*) just that it does not pop the first word (the L0 header) from the VMM Capture channel with index *VMM_Id*. Thus, there is no need to pass the *VMM_Id* argument.
next_VMM(*VMM_Id*)	Returns the index of the next associated VMM Capture channel. After the last VMM Capture channel is reached (i.e. the one with the highest index), it starts again with the first one (what first_VMM() returns).
check_header_parity (*header_data*)	Returns True if the parameter L0 header data has correct parity, False otherwise.
read_entire_L0_event (*VMM_Id*)	Pops data words from the *VMM_Id* VMM Capture channel until the beginning of the next L0 packet (it stops after the first word with a set MSB - a set MSB signals the last word of a packet).
write_trailer (*vmm_missing_bits*)	Constructs the trailer word and pushes it into the *SROC_FIFO*. The function returns only after the trailer word is successfully written (i.e. if *SROC_FIFO.full* == 1 it will wait until one address is freed and then pushes the trailer). It uses the *hit_number* global variable, the *timeout_status* of the associated VMM Capture channels and the *vmm_missing_bits* parameter. The function updates and pushes the *checksum* global variable.
write_hit_data(*VMM_Id*, *vmm_missing_bits*)	Pops all the hit words of the current L0 packet from the *VMM_Id* VMM Capture channel FIFO and transforms them into L1 hit words that are pushed into the *SROC_FIFO*. The function returns only after all the hit words are successfully written (i.e. if *SROC_FIFO.full* == 1 it will wait until at least one address is freed and then pushes the hit words, one by one). For each pushed hit word, the *hit_number* global variable is incremented and the *checksum* global variable is updated. It receives the parameter *vmm_missing_bits* and returns the updated value, with a set bit corresponding to the *VMM_Id* VMM Capture channel if any of the L0 hits have a set T bit.
write_dummy_hits (*dummy_hits_indexes*)	Generates and pushes one dummy hit word in the name of each VMM Capture channel that has a corresponding set bit flag within the *dummy_hits_indexes* parameter. The function returns only after all the dummy hit words are successfully written (i.e. if *SROC_FIFO.full* == 1 it will wait until at least one address is freed and then pushes the dummy hit words, one by one). The dummy hit words are pushed into the *SROC_FIFO* in the ascending order of the associated VMM Capture channels. If no dummy hits words must be pushed (all the *dummy_hits_indexes* are cleared) then the function simply returns.

Table A.1: Descriptions of the functions called in Algorithms 4, 5, 7 and 6.

Algorithm 4 The main algorithm implemented by the SROC Packet Builder FSM. The functions are detailed in Table A.1. The comments are detailed in Table A.2. The structures and types definitions are presented in Table A.3. The variables and constants that are not explained within the comments are described in Table A.4. The last two instructions call Algorithms 5 and 6, respectively.

```
1:  dummy_hits ← 0                                                          ▷ C1

2:  while SROC_enable do                                                    ▷ C2

3:      ▷ Initialization of variables
4:      hit_data ← False                                                    ▷ C3
5:      VMM_missing ← 0                                                     ▷ C4
6:      sel_VMM ← first_VMM()                                               ▷ C5
7:      checksum ← 0                                                        ▷ C6
8:      hit_number ← 0                                                      ▷ C7
9:      TO_bit ← 0                                                          ▷ C8

10:     while TTC_FIFO.empty do                                             ▷ C9
11:         wait_one_clock_cycle()
12:     trigger ← read_TTC_FIFO()                                           ▷ C10

13:     if bypass_mode then                                                 ▷ C11
14:         while VMM_FIFO[sel_VMM].empty do                                ▷ C12
15:             wait_one_clock_cycle()
16:         if VMM_FIFO[sel_VMM].rdata[32] == 1 then                        ▷ C13
17:             if (not VMM_FIFO[sel_VMM].rdata[31]) and
18:      VMM_FIFO[sel_VMM].rdata[27:16] != PATTERN then                     ▷ C14
19:                 if null_event_enable then                               ▷ C15
20:                     write_null_event(sel_VMM)                           ▷ C16
21:                 else
22:                     read_null_event_or_header(sel_VMM)                  ▷ C17
23:             else                                                        ▷ C18
24:                 write_header(sel_VMM)
25:                 VMM_missing ← (1 ≪ sel_VMM)
26:                 write_trailer(VMM_missing)
27:         else                                                            ▷ C19
28:             write_header(sel_VMM)
29:             VMM_missing ← write_hit_data(sel_VMM, VMM_missing)
30:             write_trailer(VMM_missing)
31:     else                                                                ▷ C20
32:         Algorithm 5
33:         Algorithm 6
```

Index	Comment
C1	Consist of eight 1-bit flags, one for each of the eight VMM Capture channels. A set flag means that a dummy hit word must be pushed into the SROC FIFO for the corresponding associated VMM Capture channel in each L1 packet. The flags persist between L1 triggers (i.e. they are not cleared at the beginning of the while loop).
C2	One iteration for each L1 trigger in the TTC FIFO resulting in one L1 event, as long as the SROC remains enabled.
C3	Flag indicating whether hit data was found for the current L1 trigger in the associated VMM Capture channels.
C4	Consists of eight 1-bit flags, one for each of the eight VMM Capture channels. A flag is set if the corresponding associated VMM Capture channel had incomplete or no data for the current L1 trigger (see Section 2.3.2 for more details). Its final value is pushed into the SROC FIFO in function write_trailer(*VMM_missing_bits*) where it is passed as the argument.
C5	The index of the currently selected associated VMM Capture channel ($sel\_VMM \in \mathbb{N},\ 0 \leqslant sel\_VMM \leqslant 7$).
C6	Global variable representing the computed checksum of the current L1 packet, updated in functions write_header(*VMM_Id*), write_header_no_read(), write_hit_data(*VMM_Id*), write_dummy_hits(*dummy_hits_indexes*) and write_trailer(*VMM_missing_bits*) - basically always when L1 data is being pushed within the SROC FIFO except for L1 null-events. Its final value is pushed into the SROC FIFO in function write_trailer(*VMM_missing_bits*).
C7	Global variable counting the number of hit words (including the dummy ones) contained in the current L1 packet, updated in functions write_hit_data(*VMM_Id*) and write_dummy_hits(*dummy_hits_indexes*). Its final value is pushed into the SROC FIFO in function write_trailer(*VMM_missing_bits*). Can be used as a flag signaling if the L1 event contains any real hit data instead of *hit_data* before the function write_dummy_hits(*dummy_hits_indexes*) is called. However, it requires a comparator circuit (e.g. a NOR gate on all its 10 bits) to reduce it to a 1-bit flag. Thus, *hit_data* is used directly as a simple 1-bit flag.
C8	The Time-Out (TO) flag is a 1-bit global variable set if at least one of the associated VMM Capture channels is in the timeout state or the SROC watchdog timer reached the configurable threshold during the construction of the current event. Its value is pushed into the SROC FIFO in function write_trailer(*VMM_missing_bits*).
C9	*TTC_FIFO.empty* is the empty signal for the TTC FIFO.
C10	Global variable of type *struct* l1_trigger_type (see Table A.3) containing the L1 trigger information based on which the L0 event selection is made. The contained values are pushed into the SROC FIFO in functions write_header(VMM_Id) and write_header_no_read().
C11	Check if the bypass feature is enabled. See Section 2.1 for more details.
C12	In this case the SROC watchdog timer does not affect the FSM operation. The SROC will wait forever for data in the associated VMM Capture FIFO.
C13	Check if the L0 packet within the associated VMM Capture FIFO is a null event (i.e. it has no hit words).

C14	Check if the L0 header's V bit is cleared and the L0 BCID is not the special *PATTERN*.
C15	Check if the sending of L1 null events is enabled for the SROC.
C16	Pop the L0 header from the associated VMM Capture FIFO, construct the corresponding L1 null event and push it into the SROC FIFO.
C17	Only pop the L0 header from the associated VMM Capture FIFO.
C18	The L0 header's V bit is set or the L0 BCID is the *PATTERN*. Both cases correspond to an L1 packet consisting of a header + trailer (with a set *VMM missing* bit for the associated VMM Capture channel).
C19	The L0 packet contains hit data. No selection based upon the BC information is done. The resulted L1 header will contain the BC information from the L0 header, not the one from *trigger*, even if the L0 BCID is the special *PATTERN*.
C20	Normal SROC operation mode.
C21	Check if the timeout feature is on and the selected VMM Capture channel is already in timeout.
C22	Clear the corresponding dummy hit flag.
C23	If all the associated VMM Capture Channels are in timeout state there is no point in iterating them one by one.
C24	Check if the last associated VMM Capture channel was processed.
C25	Check if the timeout feature is on or the watchdog timer has not reached the configurable threshold. If the timeout feature is on and this line of the algorithm was reached it means that the current selected VMM Capture channel is not in timeout. Else, the SROC watchdog timer must be checked.
C26	Check if the current header contains the *PATTERN* or indicates incorrect parity.
C27	Set the corresponding dummy hit flag.
C28	Check if the current L0 OrbitID matches the L1 trigger OrbitID.
C29	The L0 BCID is behind the L1 trigger BCID (while the OrbitIDs match) meaning that the VMM3 data is older than the L1 trigger and thus must be discarded.
C30	Check if it is a matching empty L0 header (i.e. L0 header without hit words). If this is true do not write an L1 null event into the SROC FIFO since the other associated VMM Capture channels may contain hit data.
C31	Check if at least one matching hit data word was already found for the current trigger indicating that the L1 header is already written in the SROC FIFO.
C32	This is the first L0 header containing hit data that matches the current L1 trigger so an L1 header must be written.
C33	The current L0 BCID is in front of the L1 trigger BCID.
C34	If this condition is true the current L0 header is considered in front of the L1 trigger.
C35	The current L0 header is behind the L1 trigger.
C36	The timeout feature is off but the SROC watchdog timer reached the configurable threshold.

C37	Check if at least one matching hit data word was already found for the current trigger meaning that the L1 header and at least one hit word are already written in the SROC FIFO; could also use *hit_number* for the same purpose.
C38	No matching L0 hit data was found.
C39	Check if it is necessary to send a trailer word.

Table A.2: The comments from Algorithm 4.

Type	**Definition**

l1_trigger_type	`typedef struct l1_trigger{`
	`  //the 2 LSBs of the selected OrbitID value`
	`  unsigned int orb;`
	`  //the 12-bit selected BCID value`
	`  unsigned int BCID;`
	`  //the 16 LSBs of the L1ID count value`
	`  unsigned int L1ID;`
	`} l1_trigger_type;`

fifo_type	`typedef struct word {`
	`  unsigned long int* data;`
	`  struct word* next;`
	`} word_type;`
	
	`//simple linked list of word_type instances`
	`typedef struct fifo {`
	`  //pointer to the element last pushed`
	`  word_type* front;`
	`  //pointer to the element to be poped next`
	`  word_type* back;`
	`  //the depth of the FIFO`
	`  unsigned int length;`
	
	`  //equivalent to (front == NULL)`
	`  unsigned int empty;`
	`  //equivalent to (fill_level == length)`
	`  unsigned int full;`
	`  //the number of words in the FIFO`
	`  unsigned int fill_level;`
	
	`  // = (*back).data`
	`  unsigned long int rdata;`
	`} fifo_type;`

Table A.3: The definitions of type aliases and structures used in Algorithms 4, 5, 7 and 6.

Algorithm 5 The algorithm for the L0 data selection while the SROC runs in *normal* (i.e. not in *bypass*) mode. The functions are detailed in Table A.1. The comments are detailed in Table A.2. The structures and types definitions are presented in Table A.3. The variables and constants that are not explained within the comments are described in Table A.4. This code is called at line 32 in Algorithm 4. In line 23 it calls Algorithm 7.

```
1: while True do
2:     if timeout_enable and timeout_status[sel_VMM] then                    ▷ C21
3:         TO_bit ← 1
4:         VMM_missing ← VMM_missing + (1 ≪ sel_VMM)
5:         dummy_hits ← dummy_hits & (∼ (1 ≪ sel_VMM))                       ▷ C22
6:         if all_VMMs_timeout() then                                        ▷ C23
7:             VMM_missing ← associated_VMMS
8:             write_header_no_read()
9:             break
10:        else
11:            sel_VMM ← next_VMM(sel_VMM)
12:            if sel_VMM == first_VMM() then break                          ▷ C24
13:    else
14:        if timeout_enable or (watchdog_timer <timeout_threshold) then     ▷ C25
15:            if not VMM_FIFOsel_VMM.empty then
16:                if VMM_FIFOsel_VMM.rdata[27:16] == PATTERN or
17:    (not check_header_parity(VMM_FIFOsel_VMM.rdata)) then                 ▷ C26
18:                    if VMM_FIFOsel_VMM.rdata[27:16] == PATTERN then
19:                        dummy_hits ← dummy_hits | (1 ≪ sel_VMM)           ▷ C27
20:                    read_entire_L0_event(sel_VMM)
21:                else
22:                    if trigger.orb == VMM_FIFOsel_VMM.rdata[29:28] then   ▷ C28
23:                        Algorithm 7
24:                    else
25:                        if (trigger.orb + 1) ==
26:    VMM_FIFOsel_VMM.rdata[29:28] then                                     ▷ C34
27:                            VMM_missing ← VMM_missing | (1 ≪ sel_VMM)
28:                            sel_VMM ← next_VMM(sel_VMM)
29:                            if sel_VMM == first_VMM() then break          ▷ C24
30:                        else
31:                            read_entire_L0_event(sel_VMM)                 ▷ C35
32:                            if trigger.orb ==
33:    (VMM_FIFOsel_VMM.rdata[29:28] + 1) then
34:                                dummy_hits ← dummy_hits & (∼ (1 ≪ sel_VMM)) ▷ C22
35:        else
36:            TO_bit ← 1                                                    ▷ C36
37:            VMM_missing ← VMM_missing + (1 ≪ sel_VMM)
38:            dummy_hits ← dummy_hits & (∼ (1 ≪ sel_VMM))                   ▷ C22
39:            sel_VMM ← next_VMM(sel_VMM)
40:            if sel_VMM == first_VMM() then break                          ▷ C24
```

Algorithm 6 The algorithm for the L0 data selection while the SROC runs in *normal* (i.e. not in *bypass*) mode executed after all the associated VMM Capture channels were processed for the current L1 trigger. The functions are detailed in Table A.1. The comments are detailed in Table A.2. The structures and types definitions are presented in Table A.3. The variables and constants that are not explained within the comments are described in Table A.4. This algorithm is called at line 33 in Algorithm 4.

```
1: if hit_data then                                                    ▷ C37
2:     write_dummy_hits(dummy_hits)
3:     write_trailer(VMM_missing)
4: else                                                                ▷ C38
5:     if &VMM_missing or &dummy_hits then                             ▷ C39
6:         write_header_no_read()
7:         write_dummy_hits(dummy_hits)
8:         write_trailer(VMM_missing)
9:     else
10:        if null_event_enable then write_null_event_no_read()        ▷ C15
```

Algorithm 7 The algorithm for the L0 event data selection when the SROC runs in *normal* (i.e. not in *bypass*) mode and the OrbitID of the L0 event from the selected VMM Capture channel matches with the L1 trigger OrbitID. The functions are detailed in Table A.1. The comments are detailed in Table A.2. The structures and types definitions are presented in Table A.3. The variables and constants that are not explained within the comments are described in Table A.4. This code is called at line 23 in Algorithm 5.

```
1: if VMM_FIFOsel_VMM.rdata[27:16] < trigger.BCID then                 ▷ C29
2:     read_entire_L0_event(sel_VMM)
3:     dummy_hits ← dummy_hits & (∼ (1 ≪ sel_VMM))                     ▷ C22
4: else
5:     if VMM_FIFOsel_VMM.rdata[27:16] == trigger.BCID then
6:         dummy_hits ← dummy_hits & (∼ (1 ≪ sel_VMM))                 ▷ C22
7:         if VMM_FIFOsel_VMM.rdata[32] then                           ▷ C30
8:             read_null_event_or_header(sel_VMM)
9:         else
10:            if hit_data then                                        ▷ C31
11:                read_null_event_or_header(sel_VMM)
12:            else                                                    ▷ C32
13:                hit_data ← True
14:                write_header(sel_VMM)
15:            VMM_missing ← write_hit_data(sel_VMM, VMM_missing)
16:    else                                                            ▷ C33
17:        VMM_missing ← VMM_missing | (1 ≪ sel_VMM)

18:    sel_VMM ← next_VMM(sel_VMM)
19:    if sel_VMM == first_VMM() then break                            ▷ C24
```

Name	Description
SROC_enable	Indicates the enabling state of the SROC.
null_event_enable	Indicates the enabling state for the transmission of L1 null events for the SROC.
TTC_FIFO	Instance of type fifo_type representing the TTC FIFO.
VMM_FIFO[7:0]	One-dimensional array of eight fifo_type instances associated with the eight VMM Capture FIFOs.
SROC_FIFO	Instance of type fifo_type representing the SROC FIFO.
bypass_mode	Flag indicating if the SROC operates in bypass mode.
ROC_Id	The 6-bit ROC ID number set in the digital configuration block.
PATTERN	The special L0 BCID value of 0xFE8 which signals the overflow of the L0 trigger FIFO within VMM3.
associated_VMMS	Consist of eight 1-bit flags, one for each of the eight VMM Capture channels, set only for the ones associated with the SROC.
timeout_enable	Indicates the enabling state for the timeout feature.
timeout_status[7:0]	One-dimensional array of the timeout flags for the VMM Capture channels.
watchdog_timer	SROC watchdog timer that increments when an associated channel is selected but it signals it is empty while the SROC awaits for L0 data from it. If the empty flag is dropped before the threshold is reached (i.e. the same threshold as for the timeout feature, the *timeout_threshold*), the watchdog resets.
timeout_threshold	The 9-bit threshold for the SROC and the VMM Capture watchdog timers.

Table A.4: Description of the variables and constants not explained in the comments of Algorithms 4, 5, 7 and 6.

A.2 Hardware implementation of the SROC Packet Builder

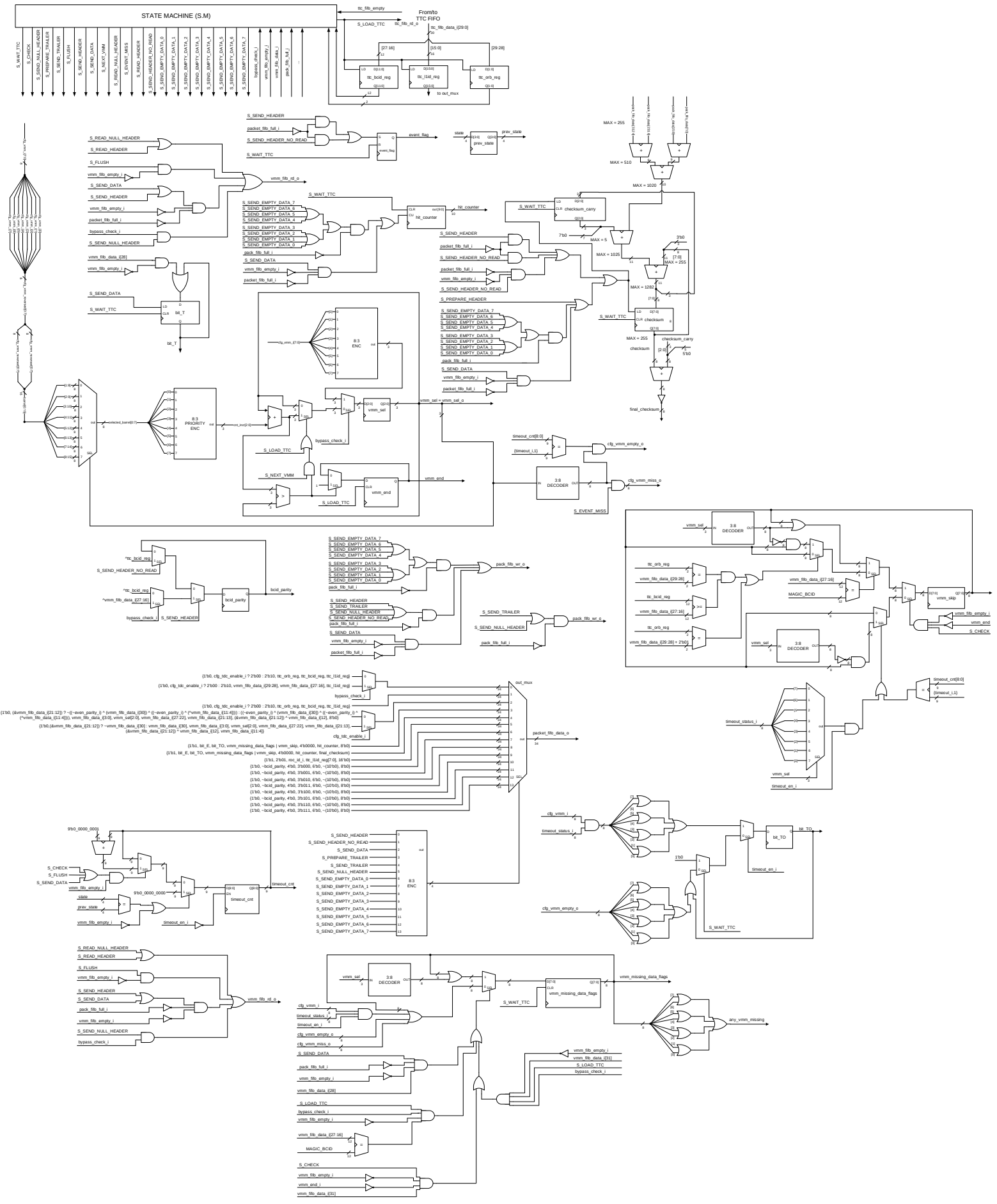
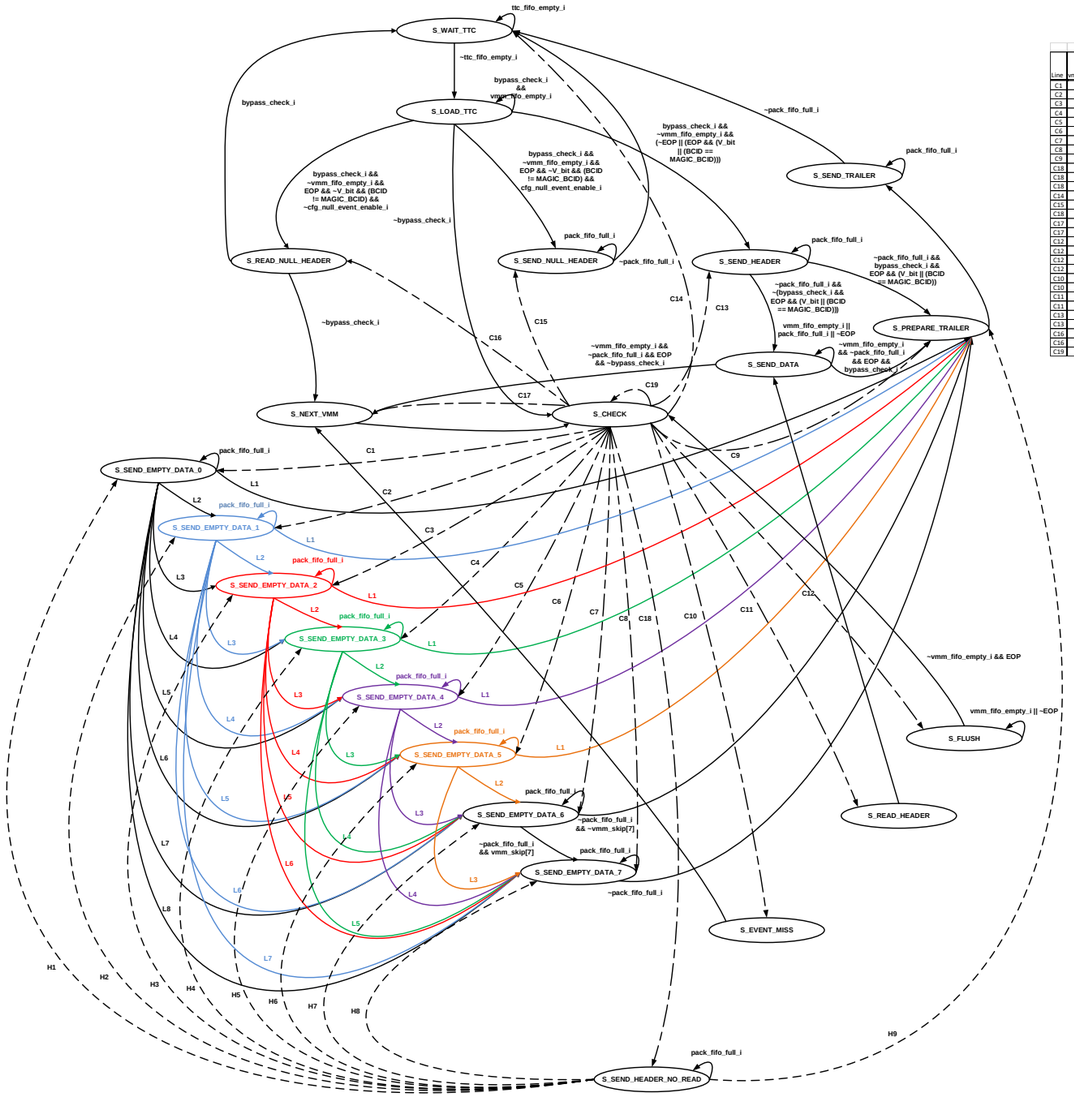


Figure A.1: The additional logic of the SROC Packet Builder FSM whose transition graph is depicted in Figure A.2.



S_CHECK:

Line	vmm_end	event_flag	vmm_skip	lvmm_skip	any_vmm_missing	cfg_null_event_enable_i	timeout_en_i	timeout_status_i[vmm_sel_i]	all the connected VMMs are in timeout	timeout_cnt >= (timeout_i, 1'b1)	vmm_fifo_empty_i	(vmm_BCID == MAGIC_BCID) || PARITY_ERROR	((ttc_orb == vmm_orb) && (vmm_bcid < ttc_bcid)) || ((ttc_orb != vmm_orb) && (ttc_orb+1 != vmm_orb))	((ttc_orb == vmm_orb) && (ttc_bcid > vmm_bcid)) || (ttc_orb + 1 == vmm_bcid)	((ttc_orb == vmm_orb) && (ttc_bcid == vmm_bcid))	EOP	Next state
C1	1	1	8'bxxxx_xxx1	1	x	x	x	x	x	x	x	x	x	x	x	x	S_SEND_EMPTY_DATA_0
C2	1	1	8'bxxxx_xx10	1	x	x	x	x	x	x	x	x	x	x	x	x	S_SEND_EMPTY_DATA_1
C3	1	1	8'bxxxx_x100	1	x	x	x	x	x	x	x	x	x	x	x	x	S_SEND_EMPTY_DATA_2
C4	1	1	8'bxxxx_1000	1	x	x	x	x	x	x	x	x	x	x	x	x	S_SEND_EMPTY_DATA_3
C5	1	1	8'bxxx1_0000	1	x	x	x	x	x	x	x	x	x	x	x	x	S_SEND_EMPTY_DATA_4
C6	1	1	8'bxx10_0000	1	x	x	x	x	x	x	x	x	x	x	x	x	S_SEND_EMPTY_DATA_5
C7	1	1	8'bx100_0000	1	x	x	x	x	x	x	x	x	x	x	x	x	S_SEND_EMPTY_DATA_6
C8	1	1	8'b1000_0000	1	x	x	x	x	x	x	x	x	x	x	x	x	S_SEND_EMPTY_DATA_7
C9	1	1	8'b0000_0000	0	x	x	x	x	x	x	x	x	x	x	x	x	S_PREPARE_TRAILER
C18	1	0	8'b0000_0000	0	1	x	x	x	x	x	x	x	x	x	x	x	S_SEND_HEADER_NO_READ
C18	1	0	at least one bit is 1	1	0	x	x	x	x	x	x	x	x	x	x	x	S_SEND_HEADER_NO_READ
C18	1	0	at least one bit is 1	1	1	x	x	x	x	x	x	x	x	x	x	x	S_SEND_HEADER_NO_READ
C14	1	0	8'b0000_0000	0	0	0	x	x	x	x	x	x	x	x	x	x	S_WAIT_TTC
C15	1	0	8'b0000_0000	0	0	1	x	x	x	x	x	x	x	x	x	x	S_SEND_NULL_HEADER
C18	0	x	x	x	x	x	1	1	1	x	x	x	x	x	x	x	S_SEND_HEADER_NO_READ
C17	0	x	x	x	x	x	1	1	0	x	x	x	x	x	x	x	S_NEXT_VMM
C17	0	x	x	x	x	x	0	x	x	1	x	x	x	x	x	x	S_NEXT_VMM
C12	0	x	x	x	x	x	0	x	x	0	0	0	x	x	x	x	S_FLUSH
C12	0	x	x	x	x	x	0	x	x	0	0	1	x	x	0	0	S_FLUSH
C12	0	x	x	x	x	x	1	0	x	x	0	0	x	x	0	0	S_FLUSH
C12	0	x	x	x	x	x	1	0	x	x	0	0	1	x	0	0	S_FLUSH
C10	0	x	x	x	x	x	0	x	x	x	0	0	0	0	1	0	S_EVENT_MISS
C10	0	x	x	x	x	x	1	0	x	x	0	0	0	1	0	0	S_EVENT_MISS
C11	0	1	x	x	x	x	0	x	x	0	0	0	0	0	1	0	S_READ_HEADER
C11	0	1	x	x	x	x	1	0	x	0	0	0	0	0	1	0	S_READ_HEADER
C13	0	0	x	x	x	x	0	x	x	0	0	0	0	0	1	0	S_SEND_HEADER
C13	0	0	x	x	x	x	1	0	0	x	0	0	0	0	1	0	S_SEND_HEADER
C16	0	x	x	x	x	x	0	x	x	0	0	0	0	0	1	1	S_READ_NULL_HEADER
C16	0	x	x	x	x	x	1	0	0	x	0	0	0	0	1	1	S_READ_NULL_HEADER
C19	ALL OTHER CONDITIONS																S_CHECK

S_SEND_EMPTY_DATA_0:

Line	packet_fifo_full_i	lvmm_skip	vmm_skip	Next state
L1	0	0	8'b0000_000x	S_PREPARE_TRAILER
L2	0	1	8'bxxxx_xx10	S_SEND_EMPTY_DATA_1
L3	0	1	8'bxxxx_x100	S_SEND_EMPTY_DATA_2
L4	0	1	8'bxxxx_1000	S_SEND_EMPTY_DATA_3
L5	0	1	8'bxxx1_0000	S_SEND_EMPTY_DATA_4
L6	0	1	8'bxx10_0000	S_SEND_EMPTY_DATA_5
L7	0	1	8'bx100_0000	S_SEND_EMPTY_DATA_6
L8	0	1	8'b1000_0000	S_SEND_EMPTY_DATA_7

S_SEND_EMPTY_DATA_1:

Line	packet_fifo_full_i	lvmm_skip	vmm_skip	Next state
L1	0	0	8'b0000_00xx	S_PREPARE_TRAILER
L2	0	1	8'bxxxx_xx10	S_SEND_EMPTY_DATA_2
L3	0	1	8'bxxxx_1000	S_SEND_EMPTY_DATA_3
L4	0	1	8'bxxx1_0000	S_SEND_EMPTY_DATA_4
L5	0	1	8'bxx10_0000	S_SEND_EMPTY_DATA_5
L6	0	1	8'bx100_0000	S_SEND_EMPTY_DATA_6
L7	0	1	8'b1000_0000	S_SEND_EMPTY_DATA_7

S_SEND_EMPTY_DATA_2:

Line	packet_fifo_full_i	lvmm_skip	vmm_skip	Next state
L1	0	0	8'b0000_0xxx	S_PREPARE_TRAILER
L2	0	1	8'bxxxx_1000	S_SEND_EMPTY_DATA_3
L3	0	1	8'bxxx1_0000	S_SEND_EMPTY_DATA_4
L4	0	1	8'bxx10_0000	S_SEND_EMPTY_DATA_5
L5	0	1	8'bx100_0000	S_SEND_EMPTY_DATA_6
L6	0	1	8'b1000_0000	S_SEND_EMPTY_DATA_7

S_SEND_HEADER_NO_READ:

Line	packet_fifo_full_i	vmm_skip	Next state
H1	0	8'bxxxx_xxx1	S_SEND_EMPTY_DATA_0
H2	0	8'bxxxx_xx10	S_SEND_EMPTY_DATA_1
H3	0	8'bxxxx_x100	S_SEND_EMPTY_DATA_2
H4	0	8'bxxxx_1000	S_SEND_EMPTY_DATA_3
H5	0	8'bxxx1_0000	S_SEND_EMPTY_DATA_4
H6	0	8'bxx10_0000	S_SEND_EMPTY_DATA_5
H7	0	8'bx100_0000	S_SEND_EMPTY_DATA_6
H8	0	8'b1000_0000	S_SEND_EMPTY_DATA_7
H9	0	8'b0000_0000	S_PREPARE_TRAILER

S_SEND_EMPTY_DATA_3:

Line	packet_fifo_full_i	lvmm_skip	vmm_skip	Next state
L1	0	0	8'b0000_xxxx	S_PREPARE_TRAILER
L2	0	1	8'bxxx1_0000	S_SEND_EMPTY_DATA_4
L3	0	1	8'bxx10_0000	S_SEND_EMPTY_DATA_5
L4	0	1	8'bx100_0000	S_SEND_EMPTY_DATA_6
L5	0	1	8'b1000_0000	S_SEND_EMPTY_DATA_7

S_SEND_EMPTY_DATA_4:

Line	packet_fifo_full_i	lvmm_skip	vmm_skip	Next state
L1	0	0	8'b000x_xxxx	S_PREPARE_TRAILER
L2	0	1	8'bxx10_0000	S_SEND_EMPTY_DATA_5
L3	0	1	8'bx100_0000	S_SEND_EMPTY_DATA_6
L4	0	1	8'b1000_0000	S_SEND_EMPTY_DATA_7

S_SEND_EMPTY_DATA_5:

Line	packet_fifo_full_i	lvmm_skip	vmm_skip	Next state
L1	0	0	8'b00xx_xxxx	S_PREPARE_TRAILER
L2	0	1	8'bx100_0000	S_SEND_EMPTY_DATA_6
L3	0	1	8'b1000_0000	S_SEND_EMPTY_DATA_7

Figure A.2: The transition graph of the SROC Packet Builder FSM except for the additional logic which is depicted in Figure A.1.

A.3 ROC IO pads

Name	Direction	Number of pads	Type	Description	Internal pulling	Internal termination	Internal configurability
vmm_clk40	Output	8	SLVS	40 MHz BC clock signals supplied to the 8 VMM3s by the analog part.	-	-	Phase control (200 ps step).
vmm_clk160	Output	8	SLVS	160 MHz RO clock signals supplied to the 8 VMM3s by the analog part.	-	-	Phase control (200 ps step).
tds_bcclk	Output	4	SLVS	40 MHz BC clock signals supplied to 4 TDSs by the analog part. Optional 160 MHz RO clock signals intended for VMM3 ART clocks.	-	-	Phase control (200 ps step).
tds_bcr	Output	4	SLVS	BCR commands to the 4 TDSs, deserialized by the TTC Capture and passed through the analog part. Optional 160 MHz RO clock signals intended for VMM3 ART clocks.	-	-	Phase control for clock signals (200 ps step), polarity control for BCRs.
eclk	Input	1	SLVS	Input reference 40 MHz BC clock signal from the L1DDC GBTx to the analog part.	-	100 Ω	No phase control within ROC.
clk160	Input	1	SLVS	Input reference 160 MHz RO clock signal as a backup in case there is a design issue with the analog part.	-	100 Ω	No phase control within ROC.
bypassForce	Input	1	Single-ended	When high the input reference clocks (i.e. *eclk* and *clk160*) bypass the ePLLs from the analog part and go directly to the digital part. This can also be done by setting the corresponding bit in the analog configuration registers.	Pull-down	-	-
pllroclocked	Output	1	Single-ended	State of the ePLL suppling BC and RO clock signals to the digital part. 0 = not locked; 1 = locked.	Pull-up	-	-

Name	Direction	Number of pads	Type	Description	Internal pulling	Internal termination	Internal configurability
plllocked	Output	1	Single-ended	AND on all the locked statuses of the ePLLs from the analog part. Individual ePLL contributions can be masked using the analog configuration registers.	Pull-up	-	-
testdown	Output	1	Single-ended	Test/debug output from the ePLL macros selectable by configuration registers.	Pull-up	-	-
testup	Output	1	Single-ended	Test/debug output from the ePLL macros selectable by configuration registers.	Pull-up	-	-
pllresetn	Input	1	Single-ended	Asynchronous active-low reset for all the ePLL macros from the analog part.	Pull-up	-	-
sresetn	Input	1	Single-ended	Asynchronous active-low reset for all the analog configuration registers. Not to be confused with the TTC SR	Pull-up	-	-
corerstn	Input	1	Single-ended	Asynchronous active-low reset for the digital part.	Pull-up	-	-
vmm_l0	Output	8	SLVS	L0A signals to the VMM3s.	-	-	Polarity control.
vmm_tp	Output	8	SLVS	TP signals to the VMM3s.	-	-	Polarity control.
vmm_bcr	Output	8	SLVS	BCR signals to the VMM3s.	-	-	Polarity control.
vmm_ena	Output	8	SLVS	Active-low SR signals to the VMM3s.	-	-	Polarity control.
scareset	Output	1	Single-ended	SCA Reset signal.	Pull-up	-	-
test_enable	Input	1	Single-ended	Debug signal that switches to parallel TTC commands, instead of the TTC stream. When 1 the parallel auxiliary inputs for SR, BCR, ECR and L1A are used instead of the TTC stream.	Pull-up	-	-

Name	Direction	Number of pads	Type	Description	Internal pulling	Internal termination	Internal configurability
test_highz	Input	1	Single-ended	Debug signal that determines the direction of the *test_l1a* and *test_ecr* pads: if high the pads are outputs, otherwise they are inputs.	Pull-down	-	-
test_l1a	Input or output	1	Single-ended	Depending on the value of *test_highz* it can be parallel auxiliary input for the L1A or debug output for the L1A obtained from the TTC stream.	Pull-up	-	-
test_ecr	Input or output	1	Single-ended	Depending on the value of *test_highz* it can be parallel auxiliary input for the ECR or debug output for the ECR obtained from the TTC stream.	Pull-up	-	-
test_bcr	Input	1	Single-ended	Parallel auxiliary input for the BCR.	Pull-up	-	-
test_softrst	Input	1	Single-ended	Parallel auxiliary input for the SR.	Pull-up	-	-
erx	Input	1	SLVS	TTC stream input from L1DDC GBTx.	-	100 Ω	-
seu	Output	1	Single-ended	Is asserted when one of the three instances of a flip-flop has another value than the other two. Covers only the digital part.	Pull-up	-	-
error	Output	1	Single-ended	OR over the error bits of the VMM Capture and SROC channels.	Pull-up	-	-
scl	Input	1	Single-ended	Clock line of the I^2C interface dedicated to the digital part configuration and status registers.	Pull-up	-	-
sda	Input or output	1	Single-ended	Data line of the I^2C interface dedicated to the digital part configuration and status registers.	Pull-up	-	-
scl2	Input	1	Single-ended	Clock line of the I^2C interface dedicated to the analog configuration and status registers.	Pull-up	-	-

Name	Direction	Number of pads	Type	Description	Internal pulling	Internal termination	Internal configurability
sda2	Input or output	1	Single-ended	Data line of the I^2C interface dedicated to the analog configuration and status registers.	Pull-up	-	-
vmm_d0	Input	8	SLVS	First lines of the L0 data lanes from VMM3	-	100 Ω	The phase is controlled by the phase of the corresponding *vmm_clk160* RO clock signal.
vmm_d1	Input	8	SLVS	Second lines of the L0 data lanes from VMM3	-	100 Ω	The phase is controlled by the phase of the corresponding *vmm_clk160* RO clock signal.
etx1	Output	4	SLVS	First data lines of the SROC L1 data lanes.	-	-	The phase is dictated by the phase of the internal RO clock signal.
etx2	Output	4	SLVS	Second data lines of the SROC L1 data lanes.	-	-	The phase is dictated by the phase of the internal RO clock signal.

Table A.5: Description of the ROC IO pads.

Acronyms

AD Antiproton Decelerator. 15
ADC Analog-to-Digital Conversion. 31, 32, 58, 59, 62, 83
ADDC ART Data Driver Card. 30, 31, 32, 36
ALICE A Large Ion Collider Experiment. 14, 16
AMBA Advanced Microcontroller Bus Architecture. 150, 159
ART Address in Real-Time. 30, 31, 32, 36, 37, 183
ASD Amplifier Shaper Discriminator. 31
ASIC Application Specific Integrated Circuit. 3, 5, 29, 30, 31, 32, 33, 34, 35, 36, 37, 39, 40, 41, 47, 48, 50, 53, 59, 60, 66, 68, 70, 73, 77, 78, 80, 81, 82, 83, 86, 89, 99, 100, 101, 102, 103, 107, 112, 113, 114, 117, 119, 121, 122, 130, 131, 145, 146, 162, 163, 164, 165, 167, 168, 169, 170, 197, 199
ATLAS A Toroidal LHC Apparatus. 3, 11, 13, 14, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 35, 36, 38, 39, 40, 63, 66, 90, 92, 132, 161, 162, 163, 164, 165, 169, 170, 195
AWAKE Advanced Proton Driven Plasma Wakefield Acceleration Experiment. 15
AXI Advanced eXtensible Interface. 90, 95, 96, 97, 150, 153, 159

BC Bunch Crossing. 16, 26, 27, 29, 31, 32, 36, 37, 38, 40, 41, 42, 43, 45, 47, 48, 50, 52, 54, 55, 56, 58, 59, 60, 61, 65, 71, 72, 89, 90, 91, 92, 93, 94, 98, 99, 100, 101, 105, 107, 118, 125, 128, 129, 141, 143, 163, 171, 175, 183
BCID BC IDentification. 29, 30, 31, 32, 36, 37, 40, 47, 48, 51, 55, 56, 58, 59, 61, 70, 76, 92, 98, 128, 129, 134, 141, 143, 144, 166, 167, 175, 178, 179
BCR BCID Counter Reset. 37, 48, 51, 56, 183, 184, 185
BGA Ball-Grid Array. 3, 39, 65, 85, 87, 88, 108, 109, 122, 196
BIST Built-In Self-Test. 80
BRAM Block BRAM. 81, 90, 150, 151, 152, 155, 158, 160, 187

CDC Clock Domain Crossing. 152, 154, 155, 166
CERN Conseil Européen pour la Recherche Nucléaire. 5, 11, 14, 15, 26, 31, 33, 82, 116, 119, 161, 164, 169, 170
CLB Configurable Logic Block. 155
CLEAR CERN Linear Electron Accelerator for Research. 15
CMOS Complementary Metal Oxide Semiconductor. 3, 33, 39, 117, 167
CMS Compact Muon Solenoid. 14, 16, 18, 27
CP Cluster Processor. 26
CPU Central Processing Unit. 66, 78, 83
CSC Cathode Strip Chamber. 20, 25
CTP Central Trigger Processor. 26, 29, 31, 36

DC Direct Current. 53

DCS Detector Control System. 30, 32
DDR Double Data Rate. 36, 38, 48, 53, 82, 83, 93, 100, 101, 119, 196
DFT Design For Testability. 80
DL Data Link. 151, 153, 158, 160
DMA Direct Memory Access. 151, 158
DoF Degrees of Freedom. 136
DRAM Dynamic RAM. 83
DRC Design Rule Check. 76, 167
DSO Digital Storage Oscilloscope. 149, 167, 168
DUT Device Under Test. 77, 81, 83, 84, 85, 99, 100, 101, 107, 108, 109, 114, 122, 148, 160, 196
DUV Device Under Verification. 78, 81, 151, 153, 154, 157, 158, 160

EC0R Event Counter L0 Reset. 37, 56, 59
ECC Error Correcting Code. 117, 118, 143, 145
ECR Event Counter Reset. 36, 37, 48, 56, 184, 185
EDA Electronic Design Automation. 162
ELENA Extra Low ENergy Antiproton. 15
EOP End Of Packet. 46, 51, 54, 58, 62, 68, 69, 70, 76, 103, 104, 126, 143
ePLL Name of the PLL from [169]. 40, 41, 50, 54, 65, 76, 82, 88, 89, 94, 98, 101, 110, 112, 117, 118, 119, 163, 166, 167, 168, 183, 184

FCal Forward Calorimeter. 23
FCC Future Circular Collider. 15
FEB Front-End Board. 30, 31, 32, 38, 53, 83
FELIX Front End LInk eXchange. 30, 31, 32, 36, 38, 41, 54, 143
FET Field Effect Transistor. 116
FF Flip-Flop. 36, 79, 80, 81, 82, 99, 115, 117, 118, 119, 120, 127, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 145, 146, 150, 151, 152, 155, 156, 158, 162, 167, 197, 199
FIFO First-In-First-Out. 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 54, 56, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 76, 83, 89, 90, 92, 93, 104, 105, 106, 118, 125, 126, 127, 128, 129, 134, 139, 140, 141, 143, 144, 145, 150, 151, 152, 154, 155, 156, 157, 158, 159, 160, 166, 168, 172, 174, 175, 176, 177, 178, 179, 195, 196, 197, 199
FMC FPGA Mezzanine Card. 85, 86, 88, 90, 122, 167
FPGA Field Programmable Gate Array. 3, 5, 30, 32, 33, 77, 81, 84, 85, 86, 88, 89, 94, 95, 98, 100, 101, 102, 103, 107, 109, 112, 113, 114, 117, 119, 120, 122, 123, 145, 147, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 160, 162, 163, 164, 167, 168, 169, 170, 197, 199
FSM Finite State Machine. 35, 38, 39, 41, 43, 44, 45, 46, 52, 58, 60, 62, 66, 71, 72, 73, 74, 76, 82, 90, 92, 93, 104, 105, 117, 125, 127, 134, 143, 148, 166, 168, 173, 174, 181, 182, 196, 197, 201
FTK Fast TracKer. 27, 28
FWFT First Word Fall Through. 50, 92, 155, 166

GBT GigaBit Transciever. 30
GBTx GigaBit Transciever. 30, 31, 32, 36, 40, 41, 53, 54, 55, 68, 83, 89, 94, 183, 185, 195

GPIO General Purpose Input/Output. 40
GTH Gigabit Transceiver grade H. 151
GUI Graphical User Interface. 149

HDL Hardware Description Language. 78, 81, 149
HEC Hadronic End-cap Calorimeter. 23
HEP High Energy Physics. 117
HIE-ISOLDE High Intensity and Energy ISOLDE. 15
HiRadMat High-Radiation to Materials. 15
HL-LHC High Luminosity LHC. 3, 28, 29, 34, 66, 161, 163
HLT High-Level Trigger. 26, 27
HPC High Pin Count. 85, 86, 88, 90, 122, 167
HVL Hardware Verification Language. 76, 78

I²C Inter-Integrated Circuit. 39, 40, 49, 50, 52, 76, 84, 88, 89, 90, 94, 95, 96, 99, 103, 110, 119, 120, 124, 167, 185, 186
IBM International Business Machines Corporation. 39
IBUFDS DIFF-OUT Input Buffer Differential Signaling with Differential Output. 94
IC Integrated Circuit. 3, 78, 79, 81, 82, 83, 86, 88, 89, 90, 116, 149, 153, 158, 160, 162
ID IDentifier. 46, 51, 59, 60, 96, 97, 129, 134, 179
IDET Inner Detector. 20, 21, 22, 23, 27, 28, 195
IEEE Institute of Electrical and Electronics Engineers. 153, 157
IFIN-HH Horia Hulubei National Institute for Research and Development in Physics and Nuclear Engineering. 5, 79, 114, 163, 164, 196
ILA Integrated Logic Analyzer. 3, 81, 85, 93, 102, 119, 122, 123, 132, 133, 134, 147, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 163, 167, 168, 170, 197
IO Input-Output. 35, 43, 47, 49, 50, 52, 64, 65, 66, 74, 75, 79, 80, 83, 84, 86, 90, 101, 112, 113, 159, 166, 167, 168, 186, 196, 199, 200
IOB Input-Output Block. 81, 94, 95, 158, 160
IP Intellectual Property. 65, 147, 149, 166
IPG Inter-Packet Gap. 157
ISOLDE Isotope Separator On Line DEvice. 15
ISR Interrupt Service Routine. 96, 97
ITk Inner Tracker. 28

JEP Jet Energy sum Processor. 26
JTAG Joint Test Action Group. 80, 81, 85, 89, 90, 122, 149, 150, 151, 159, 160

L0 Level-0. 28, 29, 31, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 51, 52, 53, 54, 56, 58, 59, 60, 61, 62, 63, 64, 66, 67, 68, 70, 71, 72, 73, 76, 82, 84, 89, 90, 92, 100, 103, 104, 120, 126, 128, 129, 130, 133, 135, 141, 142, 143, 144, 162, 166, 167, 171, 172, 174, 175, 176, 177, 178, 179, 186, 196, 201
L0A Level-0 Accept. 36, 37, 38, 41, 48, 56, 59, 62, 70, 90, 91, 184
L0ID Level-0 IDentifier. 29, 37, 59, 60
L1 Level-1. 26, 28, 29, 32, 35, 37, 38, 39, 40, 41, 42, 43, 44, 46, 47, 49, 52, 53, 54, 58, 59, 60, 61, 62, 63, 64, 66, 67, 68, 69, 70, 71, 72, 73, 76, 82, 84, 90, 92, 98,

OSI Open Systems Interconnection. 151, 153, 158, 160

PCB Printed Circuit Board. 30, 31, 32, 64, 65, 80, 84, 85, 86, 87, 88, 89, 90, 109, 112, 114, 122, 123, 145, 153, 158, 162, 163, 167, 196
PCI Peripheral Component Interconnect. 99, 191
PCIe PCI Express. 99, 151
pFEB pad FEB. 30, 32, 36, 165
Ph.D. Philosophiae Doctor. 5, 33
PHY Physical. 151, 153, 158, 160
PLD Programmable Logic Device. 149
PLL Phase-Locked Loop. 39, 40, 81, 84, 89, 100, 101, 107, 158, 160
PMR Penta Modular Redundancy. 117
PNR Place and Route. 75, 78, 83, 119
PS Proton Synchrotron. 14, 15
PSB Proton Synchrotron Booster. 14, 15, 28

QFP Quad Flat Package. 64, 65, 85, 86, 87

RAL Rutherford Appleton Laboratory. 162
RAM Random Access Memory. 36, 78, 83, 151, 152, 158, 188, 192
REX-ISOLDE Radioactive beam Experiment at ISOLDE. 15
RGB Reg-Green-Blue. 89
RIB Radioactive Ion Beam. 15
RISC Reduced Instruction Set Computer. 84, 90
RO Read-Out. 29, 36, 41, 42, 45, 47, 50, 53, 54, 55, 65, 72, 93, 94, 98, 100, 101, 118, 183, 186
ROC Read-Out Controller. 3, 5, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 56, 60, 61, 62, 63, 64, 65, 66, 67, 68, 70, 72, 73, 74, 76, 77, 78, 79, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 129, 130, 131, 132, 133, 134, 143, 144, 145, 146, 147, 150, 151, 152, 155, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 179, 183, 186, 195, 196, 197, 199, 200, 201
ROD Read-Out Driver. 26, 32
ROI Region Of Interest. 24, 26, 29, 32, 62
ROS Read-Out System. 26
RPC Resistive Plate Chamber. 20, 25, 26
RTL Register Transfer Level. 64, 74, 75, 76, 78, 81, 92, 93, 102, 107, 119, 145, 149, 166, 167

SCA Slow Control Adapter. 30, 31, 32, 37, 39, 40, 41, 48, 56, 165, 184, 197
SCL Serial Clock Line. 50, 88, 119, 120
SDA Serial Data Line. 50, 88, 119, 120
SDC Synopsys Design Constraints. 74, 75, 83, 119, 167
SDF Standard Delay Format. 74, 75, 76, 78, 99, 109, 163, 167
SEB Single Event Burnout. 116
SEE Single Event Effect. 115, 116, 117
SEGR Single Event Gate Rupture. 116
SEL Single Event Latchup. 116

SET Single Event Transient. 115, 116, 118
SEU Single Event Upset. 3, 39, 52, 58, 61, 65, 79, 115, 116, 118, 119, 120, 124, 125, 127, 130, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 144, 145, 146, 150, 166, 197, 199, 200
sFEB strip FEB. 30, 32, 36, 165
SFP+ Small Form-factor Pluggable. 151
SI Système International (d'unités). 116
SLVS Scalable Low-Voltage Signaling. 36, 38, 51, 53, 65, 86, 88, 183, 184, 185, 186
SM Standard Model. 12, 17, 18, 161, 195
SMA SubMiniature version A. 88
SNR Signal-to-Noise Ratio. 31
SOC System On a Chip. 89
SOP Start Of Packet. 54, 58, 62, 68, 104, 126, 143
SPI Serial Peripheral Interface. 41, 99
SPICE Simulation Program with Integrated Circuit Emphasis. 78, 79
SPS Super Proton Synchroton. 14, 15
SR Soft-Rest. 36, 41, 48, 49, 50, 56, 142, 143, 184, 185
SRAM Static RAM. 36, 38, 50, 58, 63, 64, 65, 75, 76, 81, 82, 98, 117, 118, 125, 126, 127, 130, 133, 134, 136, 137, 138, 141, 142, 143, 145, 146, 150, 163, 166, 168, 197, 199, 200
SROC Sub-ROC. 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 54, 56, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 82, 84, 89, 90, 92, 93, 94, 97, 98, 103, 104, 105, 106, 107, 110, 111, 112, 123, 125, 126, 127, 128, 129, 133, 134, 135, 139, 141, 142, 143, 144, 145, 146, 155, 166, 171, 172, 173, 174, 175, 176, 177, 178, 179, 181, 182, 185, 186, 195, 196, 197, 199, 201
STA Static Timing Analysis. 74, 75, 79, 83, 99
sTGC small-strip Thin Gap Chamber. 28, 29, 30, 31, 32, 36, 59, 107, 163, 165, 197
SUSY supersymmetry. 18, 161
SVA SystemVerilog Assertion. 81

TCL Tool Command Language. 123, 149
TD Testing Device. 77
TDAQ Trigger and Data Acquisition. 3, 11, 26, 27, 28, 29, 30, 32, 34, 35, 38, 39, 56, 58, 60, 66, 89, 92, 111, 112, 129, 161, 164, 168, 195
TDC Time to Digital Converter. 46, 51, 58, 59, 60, 62, 70, 103
TDS Trigger Data Serializer. 30, 31, 32, 36, 37, 47, 48, 183
TGC Thin Gap Chamber. 20, 25, 26
TID Total Ionizing Dose. 115, 116, 120, 121, 196
TMR Triple Modular Redundancy. 39, 41, 100, 117, 118, 119, 127, 143, 145, 163, 167, 196
ToT Time-over-Threshold. 32
TP Test Pulse. 32, 37, 48, 56, 184
TPROC Trigger Processor. 32
TT Transfer Tunnel. 15
TTC Time Trigger and Control. 29, 31, 32, 36, 37, 39, 40, 41, 43, 45, 46, 47, 48, 49, 50, 51, 52, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 67, 71, 72, 77, 82, 83, 84, 86, 88, 89, 90, 92, 93, 94, 97, 98, 100, 101, 102, 103, 105, 106, 110, 111, 112, 118, 119, 123, 126, 127, 128, 129, 141, 142, 143, 144, 145, 162, 166, 174, 179, 183, 184, 185, 195, 196, 197, 199

UART Universal Asynchronous Receiver-Transmitter. 89, 90, 95, 96, 122, 123
USA15 Underground Service ATLAS. 31
USB Universal Serial Bus. 99, 122, 149, 150, 151, 159, 160

VE Verification Environment. 74, 75, 76, 78, 81, 82, 151, 153, 154, 157, 166, 167
VITA Versa Module Europa or Versa Module Eurocard bus (VMEbus) International Trade Association. 85, 122
VMEbus Versa Module Europa or Versa Module Eurocard bus. 85, 193
VMM Venetios Micro-Megas. 32
VTRx Versatile Transceiver. 31

WRR Weighted Round Robin. 153, 156
WWW World Wide Web. 11

XAUI 10 Gigabit Attachment Unit Interface. 153, 158
XGMII 10 Gigabit Media-Independent Interface. 153, 158, 160

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] Address in real-time concentrator asic. Available at https://espace.cern.ch/ATLAS-NSW-ELX/Shared%20Documents/ADDC/ART%20ASIC/artasic_specs1.2_020316.pdf. Accessed: 02-04-2021.

[2] Atlas authorship policy. version 7.4. Available at https://twiki.cern.ch/twiki/pub/AtlasProtected/AtlasPolicyDocuments/Authorship_Policy.pdf. Accessed: 26-05-2021.

[3] Atmega16. 8-bit avr microcontroller with 16k bytes in-system programmable flash. Technical report. Accessed: 20 May 2018. URL: https://ww1.microchip.com/downloads/en/DeviceDoc/doc2466.pdf.

[4] The birth of the web. Available at https://home.cern/science/computing/birth-web. Accessed: 17-03-2021.

[5] The cern accelerator complex - 2019 complexe des accélérateurs du cern - 2019. Available at https://cds.cern.ch/record/2684277/. Accessed: 22-03-2021.

[6] First beam in the lhc - accelerating science. Available at https://home.cern/news/press-release/cern/first-beam-lhc-accelerating-science. Accessed: 26-03-2021.

[7] Roc digital documentation. Available at https://espace.cern.ch/ATLAS-NSW-ELX/Shared%20Documents/ROC/ROC_digital_documentation.pdf. Co-authored by the author.

[8] Super proton synchrotron marks its 25th birthday. Available at https://cerncourier.com/a/super-proton-synchrotron-marks-its-25th-birthday/. Accessed: 23-03-2021.

[9] First test of smallest "atom smasher". *Electrical Engineering*, 75(2):208–209, 1956. doi:10.1109/EE.1956.6442465.

[10] Isolde isotope separator on-line project. *CERN Courier Volume 7, Number 2*, pages 23–27, feb 1967. URL: https://cds.cern.ch/record/1728851/files/vol7-issue2.pdf.

[11] *Particle Accelerators*, chapter 12, pages 363–393. Springer, Boston, MA, USA, 1972. doi:10.1007/978-1-4615-9701-8_12.

[12] A new linac - un noveau linac. *CERN Bulletin Issue No. 46/1973*, page 1, nov 1973. URL: https://cds.cern.ch/record/1716524.

[13] New linac + 'old' booster = many protons - nouveau linac + 'booster' = multiplication des protons. *CERN Bulletin Issue No. 45/1978*, pages 1–2, nov 1978. URL: https://cds.cern.ch/record/1718567/files/45-1978-p001.pdf.

[14] Leading lead ions towards physics, first full acceleration of ions in the lead linac - vers l'expérimentation, premièr e pleine accélération des ions dans le linac à ions plom b. *CERN Bulletin Issue No. 24/1994*, pages 1–3, june 1994. URL: https://cds.cern.ch/record/1725529/files/24-1994.pdf.

[15] *ATLAS calorimeter performance: Technical Design Report*. Technical design report. ATLAS. CERN, Geneva, 1996. URL: https://cds.cern.ch/record/331059.

[16] *ATLAS liquid-argon calorimeter: Technical Design Report*. Technical design report. ATLAS. CERN, Geneva, 1996. URL: https://cds.cern.ch/record/331061.

[17] *ATLAS inner detector: Technical Design Report, 1*. Technical design report. ATLAS. CERN, Geneva, 1997. URL: http://cds.cern.ch/record/331063.

[18] *ATLAS magnet system: Technical Design Report, 1.* Technical design report. ATLAS. CERN, Geneva, 1997. URL: https://cds.cern.ch/record/338080.

[19] *ATLAS muon spectrometer: Technical Design Report.* Technical design report. ATLAS. CERN, Geneva, 1997. URL: https://cds.cern.ch/record/331068.

[20] Asic design guidelines. Technical report, December 1999. Accessed: 20 May 2017. URL: http://157.158.56.13/Electronics_Firm_Docs/ATMEL/Atmel/acrobat/doc1205.pdf.

[21] *Coding Techniques*, chapter 2, pages 30–82. John Wiley & Sons, Ltd, 2002. doi:10.1002/047122460X.ch2.

[22] *The American Heritage Science Dictionary.* Houghton Mifflin Harcourt, Boston, Massachusetts, USA, 2005. doi:10.1007/978-94-007-2464-8.

[23] Ieee standard for test access port and boundary-scan architecture. *IEEE Std 1149.1-2013 (Revision of IEEE Std 1149.1-2001)*, pages 1–444, May 2013. doi:10.1109/IEEESTD.2013.6515989.

[24] Chapter 6 - the case for synchronous design. In Hubert Kaeslin, editor, *Top-Down Digital VLSI Design*, pages 357–389. Morgan Kaufmann, Boston, 2015. doi:https://doi.org/10.1016/B978-0-12-800730-3.00006-X.

[25] Chapter 7 - clocking of synchronous circuits. In Hubert Kaeslin, editor, *Top-Down Digital VLSI Design*, pages 391–443. Morgan Kaufmann, Boston, 2015. doi:https://doi.org/10.1016/B978-0-12-800730-3.00007-1.

[26] *Latchup*, chapter 11, pages 206–229. John Wiley & Sons, Ltd, 2016. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118707128.ch11, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118707128.ch11, doi:https://doi.org/10.1002/9781118707128.ch11.

[27] ATLAS Liquid Argon Calorimeter Phase-II Upgrade: Technical Design Report. Technical report, CERN, Geneva, Sep 2017. URL: https://cds.cern.ch/record/2285582.

[28] Technical Design Report for the ATLAS Inner Tracker Pixel Detector. Technical report, CERN, Geneva, Sep 2017. URL: https://cds.cern.ch/record/2285585.

[29] Technical Design Report for the Phase-II Upgrade of the ATLAS Muon Spectrometer. Technical report, CERN, Geneva, Sep 2017. URL: https://cds.cern.ch/record/2285580.

[30] Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System. Technical report, CERN, Geneva, Sep 2017. URL: https://cds.cern.ch/record/2285584.

[31] Technical Design Report for the Phase-II Upgrade of the ATLAS Tile Calorimeter. Technical report, CERN, Geneva, Sep 2017. URL: https://cds.cern.ch/record/2285583.

[32] Ieee standard for ethernet. *IEEE Std 802.3-2018 (Revision of IEEE Std 802.3-2015)*, pages 1–5600, Aug 2018. doi:10.1109/IEEESTD.2018.8457469.

[33] Ieee standard for systemverilog–unified hardware design, specification, and verification language. *IEEE Std 1800-2017 (Revision of IEEE Std 1800-2012)*, pages 1–1315, Feb 2018. doi:10.1109/IEEESTD.2018.8299595.

[34] Technical Design Report: A High-Granularity Timing Detector for the ATLAS Phase-II Upgrade. Technical report, CERN, Geneva, Jun 2020. URL: http://cds.cern.ch/record/2719855.

[35] Gbtx manual v0.17. Available at https://espace.cern.ch/GBT-Project/GBTX/Manuals/gbtxManual.pdf, March 2021. Accessed: 09-04-2021.

[36] G. Aad et al. The ATLAS Inner Detector commissioning and calibration. *Eur. Phys. J. C*, 70:787–821, 2010. arXiv:1004.5293, doi:10.1140/epjc/s10052-010-1366-7.

[37] A et al. Abada. He-lhc: The high-energy large hadron collider. *The European Physical Journal Special Topics*, 228(5):1109–1382, Jul 2019. doi:10.1140/epjst/e2019-900088-6.

[38] European Space Agency. Single event effects test method and guidelines., Oct 2014. ESCC Basic Specification No. 25100, Accessed: 15 June 2020. URL: https://escies.org/download/webDocumentFile?id=62690.

[39] M Aleksa and M Diemoz. Discussion on the electromagnetic calorimeters of ATLAS and CMS. Technical report, CERN, Geneva, May 2013. URL: https://cds.cern.ch/record/1547314.

[40] R Alemany-Fernandez, E Bravin, L Drosdal, and A et al. Gorzawski. Operation and Configuration of the LHC in Run 1. Nov 2013. URL: https://cds.cern.ch/record/1631030.

[41] T. Alexopoulos, G. Fanourakis, T. Geralis, M. Kokkoris, A. Kourkoumeli-Charalampidi, K. Papageorgiou, and G. Tsipolitis. Study of the VMM1 read-out chip in a neutron irradiation environment. *Journal of Instrumentation*, 11(05):P05015–P05015, may 2016. doi:10.1088/1748-0221/11/05/p05015.

[42] Arnold O. Allen. *Probability, Statistics, and Queueing Theory With Computer Science Applications*. Computer Science and Scientific Computing. Academic Press, INC., San Diego, CA, USA, 1990.

[43] M. Allenspach, J.R. Brews, K.F. Galloway, G.H. Johnson, R.D. Schrimpf, R.L. Pease, J.L. Titus, and C.F. Wheatley. Segr: A unique failure mode for power mosfets in spacecraft. *Microelectronics Reliability*, 36(11):1871–1874, 1996. Reliability of Electron Devices, Failure Physics and Analysis. doi:https://doi.org/10.1016/0026-2714(96)00218-1.

[44] I. Béjar Alonso, O. Brüning, P. Fessia, M. Lamont, L. Rossi, L. Tavian, and M. Zerlauth. Vol. 10 (2020): High-luminosity large hadron collider (hl-lhc): Technical design report. Cern yellow reports: Monographs, CERN, 2020. URL: https://e-publishing.cern.ch/index.php/CYRM/issue/view/127/95.

[45] Altera. Design debugging using the signaltap ii logic analyzer. Available at https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/qts/qts_qii53009.pdf, 2013. Accessed: 19-04-2021.

[46] Altera. Recommended design practices, quartus ii handbook. Available at https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/qts/qts_qii51006.pdf, November 2013. Accessed: 19-04-2018.

[47] L Amaral, S Dris, A Gerardin, T Huffman, C Issever, A J Pacheco, M Jones, S Kwan, S C Lee, Z Liang, T Liu, Z Meng, A Prosser, S Padadopoulos, I Papakonstanstinou, C Sigaud, S Silva, C Soos, P Stejskal, J Troska, F Vasey, P Vichoudis, T Weidberg, A Xiang, and J Ye. The versatile link, a common project for super-LHC. *Journal of Instrumentation*, 4(12):P12003–P12003, dec 2009. doi:10.1088/1748-0221/4/12/p12003.

[48] Dag Gillberg and. Performance of the ATLAS forward calorimeters in first LHC data. *Journal of Physics: Conference Series*, 293:012041, apr 2011. doi:10.1088/1742-6596/293/1/012041.

[49] J Anderson, A Borga, H Boterenbrood, H Chen, K Chen, G Drake, D Francis, B Gorini, F Lanni, G Lehmann Miotto, L Levinson, J Narevicius, C Plessl, A Roich, S Ryu, F Schreuder, J Schumacher, W Vandelli, J Vermeulen, and J Zhang. FELIX: a high-throughput network approach for interfacing to front end electronics for ATLAS upgrades. *Journal of Physics: Conference Series*, 664(8):082050, dec 2015. doi:10.1088/1742-6596/664/8/082050.

[50] ANSI/VITA. Vita 57 fpga mezzanine card (fmc). signals and pinout of high-pin count (hpc) and low-pin count (lpc) connectors. Available at https://fmchub.github.io/appendix/VITA57_FMC_HPC_LPC_SIGNALS_AND_PINOUT.html. Accessed: 19-02-2021.

[51] ARM. Amba 4 axi4-stream protocol, version: 1.0, specification. Available at https://developer.arm.com/documentation/ihi0051/a/Introduction/About-the-AXI4-Stream-protocol, 2010. Accessed: 19-04-2021.

[52] L. Artola, M. Gaillardin, G. Hubert, M. Raine, and P. Paillet. Modeling single event transients in advanced devices and ics. *IEEE Transactions on Nuclear Science*, 62(4):1528–1539, Aug 2015. doi:10.1109/TNS.2015.2432271.

[53] S Ask, D Berge, P Borrego-Amaral, D Caracinha, N Ellis, P Farthouat, P Gällnö, S Haas, J Haller, P Klofver, A Krasznahorkay, A Messina, C Ohm, T Pauly, M Perantoni, H Pessoa Lima Junior, G Schuler, D Sherman, R Spiwoks, T Wengler, J M de Seixas, and R Torga Teixeira. The ATLAS central level-1 trigger logic and TTC system. *Journal of Instrumentation*, 3(08):P08002–P08002, aug 2008. doi:10.1088/1748-0221/3/08/p08002.

[54] Solid State Technology Association. Scalable low-voltage signaling for 400 mv (slvs-400). Standard jesd8-13, JEDEC, Oct 2001. URL: http://www.jedec.org/sites/default/files/docs/jesd8-13.pdf.

[55] R. Bailey, editor. *CAS - CERN Accelerator School: Power Converters - Radiation Risks and Mitigation in Electronic Systems*, Geneva, May 2014. CERN, CERN. URL: https://cds.cern.ch/record/2038628/files/245-263-Todd.pdf, doi:10.5170/CERN-2015-003.

[56] Moises Barbera Ramos. LHC TI2/TI8 TL Stability Check RUN II . TL - Transfer Line. Sep 2019. URL: http://cds.cern.ch/record/2688624.

[57] F. Bauer, U. Bratzler, H. Dietl, H. Kroha, Th. Lagouri, A. Manz, A. Ostapchuk, R. Richter, S. Schael, S. Chouridou, M. Deile, O. Kortner, A. Staude, R. Ströhmer, and T. Trefzger. Construction and test of mdt chambers for the atlas muon spectrometer. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 461(1):17–20, 2001. 8th Pisa Meeting on Advanced Detectors. doi:https://doi.org/10.1016/S0168-9002(00)01156-6.

[58] P. Belochitskii, J. Bosser, and J. Buttkus et al. Commissioning and first operation of the antiproton decelerator (ad). In *PACS2001. Proceedings of the 2001 Particle Accelerator Conference (Cat. No.01CH37268)*, volume 1, pages 580–584 vol.1, 2001. doi:10.1109/PAC.2001.987574.

[59] P. Beloshitsky et al. LEIR commissioning. *Conf. Proc. C*, 060626:1876–1878, 2006. URL: https://cds.cern.ch/record/972341/files/lhc-project-report-923.pdf.

[60] Evert Birgersson and Göran Lövestam. Neusdesc – neutron source description software manual. Technical report, 2009. Accessed: 15 June 2020. URL: http://publications.jrc.ec.europa.eu/repository/bitstream/JRC51437/reqno_jrc51437_jrc51437-eur_number_23794[1].pdf.

[61] ATLAS Resources Review Board. Atlas upgrade status report 2018 - 2019. Technical report, October 2018. Accessed: 29 October 2019. URL: https://cds.cern.ch/record/2638085/files/CERN-RRB-2018-079.pdf.

[62] S Bonacini, K Kloukinas, and P Moreira. E-link: A Radiation-Hard Low-Power Electrical Link for Chip-to-Chip Communication. 2009. URL: https://cds.cern.ch/record/1235849, doi:10.5170/CERN-2009-006.422.

[63] Catalin Borcea, S. Buono, P Cennini, and Marcus et al. Dahlfors. First results from the neutron facility (ntof) at cern. *Applied Physics A*, 74, 12 2002. doi:10.1007/s003390201610.

[64] F Bordry, S Baird, K Foraz, A L Perrot, R Saban, and J Ph Tock. The First Long Shutdown (LS1) for the LHC. *Conf. Proc.*, C130512(CERN-ACC-2013-0084):MOZB202. 6 p, Jul 2013. URL: https://cds.cern.ch/record/1575133.

[65] M.J.G. Borge. Highlights of the isolde facility and the hie-isolde project. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 376:408–412, 2016. Proceedings of the XVIIth International Conference on Electromagnetic Isotope Separators and Related Topics (EMIS2015), Grand Rapids, MI, U.S.A., 11-15 May 2015. doi:https://doi.org/10.1016/j.nimb.2015.12.048.

[66] N. Bourbaki. *Elements of Mathematics. Theory of Sets*. Springer-Verlag, 3rd edition, 2004.

[67] Sylvie Braibant and Giorgio Giacomelli. *Particles and fundamental interactions: An introduction to particle physics*. Undergraduate lecture notes in physics. Springer, Dordrecht, Netherlands, 2012. doi:10.1007/978-94-007-2464-8.

[68] Xilinx Brandon Jiao. Leveraging ultrascale architecture transceivers for high-speed serial i/o connectivity. Available at https://www.xilinx.com/support/documentation/white_papers/wp458-ultrascale-xcvrs-serialio.pdf, October 2019. Accessed: 19-04-2021.

[69] P J Bryant. A brief history and review of accelerators. 1994. URL: http://cds.cern.ch/record/261062, doi:10.5170/CERN-1994-001.1.

[70] Jean-Paul Burnet, Christian Carli, and Michel et al. Chanel. *Fifty years of the CERN Proton Synchrotron: Volume 1*. CERN Yellow Reports: Monographs. CERN, Geneva, 2011. URL: https://cds.cern.ch/record/1359959, doi:10.5170/CERN-2011-004.

[71] Laurent Canetti, Marco Drewes, and Mikhail Shaposhnikov. Matter and antimatter in the universe. *New Journal of Physics*, 14(9):095012, sep 2012. doi:10.1088/1367-2630/14/9/095012.

[72] S. Carroll and Teaching Company. *Dark Matter, Dark Energy: The Dark Side of the Universe*. Great courses. Teaching Company, LLC, 2007. URL: https://books.google.ro/books?id=oSGEXwAACAAJ, doi:10.1063/1.2337829.

[73] R Catherall, W Andreazza, and M Breitenfeldt et al. The ISOLDE facility. *Journal of Physics G: Nuclear and Particle Physics*, 44(9):094002, aug 2017. doi:10.1088/1361-6471/aa7eba.

[74] CERN. Lhc season 2. facts & figures. Available at https://home.cern/sites/home.web.cern.ch/files/2018-07/CERN-Brochure-2015-003-Eng.pdf. Accessed: 05-01-2021.

[75] CERN. Lhc the guide. Available at https://home.cern/sites/home.web.cern.ch/files/2018-07/CERN-Brochure-2017-002-Eng.pdf. Accessed: 05-01-2021.

[76] N. Charitonidis, A. Fabich, and I. Efthymiopoulos. Hiradmat: A high-energy, pulsed beam, material irradiation facility. In *2015 4th International Conference on Advancements in Nuclear Instrumentation Measurement Methods and their Applications (ANIMMA)*, pages 1–3, April 2015. doi:10.1109/ANIMMA.2015.7465596.

[77] John Douglas Cockcroft and E. T. S. Walton. Experiments with high velocity positive ions. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 129(811):477–489, 1930. doi:10.1098/rspa.1930.0169.

[78] John Douglas Cockcroft, E. T. S. Walton, and Ernest Rutherford. Experiments with high velocity positive ions. ii. -the disintegration of elements by high velocity protons. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 137(831):229–242, 1932. doi:10.1098/rspa.1932.0133.

[79] R.-M. Coliban, S. Popa, T. Tulbure, D. Nicula, M. Ivanovici, S. Martoiu, L. Levinson, and J. Vermeulen. The read out controller for the ATLAS new small wheel. *Journal of Instrumentation*, 11(02):C02069–C02069, feb 2016. doi:10.1088/1748-0221/11/02/c02069.

[80] ATLAS Collaboration. Search for flavour-changing neutral currents in processes with one top quark and a photon using 81 fb(-1) of pp collisions at root s=13 TeV with the ATLAS experiment. *PHYSICS LETTERS B*, 800, JAN 10 2020.

[81] ATLAS collaboration. Performance of the ATLAS RPC detector and Level-1 muon barrel trigger at $\sqrt{s} = 13$ TeV. 2 2021. URL: http://cds.cern.ch/record/2753039/files/ANA-MDET-2019-01-PAPER.pdf?version=1, arXiv:2103.01029.

[82] The ATLAS collaboration. ATLAS pixel detector electronics and sensors. *Journal of Instrumentation*, 3(07):P07007–P07007, jul 2008. doi:10.1088/1748-0221/3/07/p07007.

[83] The ATLAS Collaboration. Commissioning of the atlas muon spectrometer with cosmic rays. *The European Physical Journal C*, 70(3):875–916, Dec 2010. doi:10.1140/epjc/s10052-010-1415-2.

[84] The ATLAS collaboration. Operation and performance of the ATLAS semiconductor tracker. *Journal of Instrumentation*, 9(08):P08009–P08009, aug 2014. doi:10.1088/1748-0221/9/08/p08009.

[85] The ATLAS Collaboration. Performance of the atlas trigger system in 2015. *The European Physical Journal C*, 77(5):317, May 2017. doi:10.1140/epjc/s10052-017-4852-3.

[86] The ATLAS collaboration. Production and integration of the ATLAS insertable b-layer. *Journal of Instrumentation*, 13(05):T05008–T05008, may 2018. doi:10.1088/1748-0221/13/05/t05008.

[87] N. Colonna, A. Tsinganis, and R. et al. Vlastou. The fission experimental programme at the cern n_tof facility: status and perspectives. *The European Physical Journal A*, 56(2):48, Feb 2020. doi:10.1140/epja/s10050-020-00037-8.

[88] J. W. Coltman. The westinghouse atom smasher???an ieee historical milestone. *IEEE Transactions on Education*, E-30(1):37–42, 1987. doi:10.1109/TE.1987.5570584.

[89] Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, and Philips. Universal serial bus specification 2.0. Available at https://www.pjrc.com/teensy/beta/usb20.pdf, April 2000. Accessed: 15-05-2017.

[90] C. Cummings. Simulation and Synthesis Techniques for Asynchronous FIFO Design. In *Proc. Synopsys User Group Meeting (SNUG)*, San Jose, CA, USA, Apr. 2002. URL: http://www.sunburst-design.com/papers/CummingsSNUG2002SJ_FIFO1.pdf.

[91] C. Cummings. Clock domain crossing (cdc) design & verification techniques using systemverilog. In *Proc. Synopsys User Group Meeting (SNUG)*, Boston, MA, USA, Sep. 2008. URL: http://www.sunburst-design.com/papers/CummingsSNUG2008Boston_CDC.pdf.

[92] C. Cummings and P. Alfke. Simulation and Synthesis Techniques for Asynchronous FIFO Design with Asynchronous Pointer Comparisons. In *SNUG*, San Jose, CA, USA, Apr. 2002. URL: http://www.sunburst-design.com/papers/CummingsSNUG2002SJ_FIFO2.pdf.

[93] C. Cummings and D. Mills. Synchronous Resets? Asynchronous Resets? I am so confused! How will I ever know which to use? In *SNUG*, San Jose, CA, USA, Apr. 2002. URL: http://www.sunburst-design.com/papers/CummingsSNUG2002SJ_Resets.pdf.

[94] G. De Geronimo, J. Fried, S. Li, J. Metcalfe, N. Nambiar, E. Vernon, and V. Polychronakos. Vmm1 - an asic for micropattern detectors. In *2012 IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC)*, pages 633–639, Oct 2012. doi:10.1109/NSSMIC.2012.6551184.

[95] Michael Dine. *Supersymmetry and String Theory. Beyond the Standard Model*. Cambridge University Press, New York, 2nd edition, 2016.

[96] A. Einstein. *Relativity: The Special and General Theory*. Read Books, 2009. URL: https://books.google.ro/books?id=x49nkF7HYncC.

[97] Albert Einstein. Über einen die erzeugung und verwandlung des lichtes betreffenden heuristischen gesichtspunkt - concerning an heuristic point of view toward the emission and transformation of light. *Annalen der Physik*, 17:132–148, 1905. URL: https://people.isy.liu.se/jalar/kurser/QF/references/Einstein1905b.pdf, doi:https://doi.org/10.1002/andp.19053220607.

[98] A. Abusleme et al. Performance of a full-size small-strip thin gap chamber prototype for the atlas new small wheel muon upgrade. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 817:85–92, 2016. doi:10.1016/j.nima.2016.01.087.

[99] ATLAS Collaboration et al. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, 716(1):1–29, 2012. doi:https://doi.org/10.1016/j.physletb.2012.08.020.

[100] CMS Collaboration et al. Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc. *Physics Letters B*, 716(1):30–61, 2012. doi:https://doi.org/10.1016/j.physletb.2012.08.021.

[101] The ALICE Collaboration et al. The ALICE experiment at the CERN LHC. *Journal of Instrumentation*, 3(08):S08002–S08002, aug 2008. doi:10.1088/1748-0221/3/08/s08002.

[102] The ALPHA Collaboration et al. Confinement of antihydrogen for 1,000 seconds. *Nature Physics*, 7(7):558–564, Jul 2011. doi:10.1038/nphys2025.

[103] The ATLAS Collaboration et al. The ATLAS experiment at the CERN large hadron collider. *Journal of Instrumentation*, 3(08):S08003–S08003, aug 2008. doi:10.1088/1748-0221/3/08/s08003.

[104] The CMS Collaboration et al. The CMS experiment at the CERN LHC. *Journal of Instrumentation*, 3(08):S08004–S08004, aug 2008. doi:10.1088/1748-0221/3/08/s08004.

[105] The LHCb Collaboration et al. The LHCb detector at the LHC. *Journal of Instrumentation*, 3(08):S08005–S08005, aug 2008. doi:10.1088/1748-0221/3/08/s08005.

[106] Lyndon Evans and Philip Bryant. LHC machine. *Journal of Instrumentation*, 3(08):S08001–S08001, aug 2008. doi:10.1088/1748-0221/3/08/s08001.

[107] Lyndon Evans and Lyn Evans. *The Large Hadron Collider: a marvel of technology; 2nd ed.* Physics (EPFL Press). EPFL Press, Lausanne, 2018. On the cover : Including the discovery of the higgs boson. URL: https://cds.cern.ch/record/2645935.

[108] Philippe Farthouat. Introduction to electronics in hep experiments. Available at https://indico.cern.ch/event/57487/attachments/989598/1407137/Summer-students-2009-part1.ppt. Accessed: 02-04-2021.

[109] Philippe Farthouat. New small wheels roc production readiness review. Technical report, May 2018. Accessed: 19 May 2018. URL: https://edms.cern.ch/ui/file/1977683/1/ROC-PRR_docx_cpdf.pdf.

[110] T. Flick. The phase II ATLAS pixel upgrade: the inner tracker (ITk). *Journal of Instrumentation*, 12(01):C01098–C01098, jan 2017. doi:10.1088/1748-0221/12/01/c01098.

[111] Paolo Francavilla. The ATLAS tile hadronic calorimeter performance at the LHC. *Journal of Physics: Conference Series*, 404:012007, dec 2012. doi:10.1088/1742-6596/404/1/012007.

[112] Giovanni Franchi. Fundamentals of electronics for high energy physics. Available at https://indico.cern.ch/event/566138/contributions/2287560/attachments/1400219/2139194/FUNDAMENTALS_r2.pdf. Accessed: 02-04-2021.

[113] J. J. Thomson M.A. F.R.S. Xl. cathode rays. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 44(269):293–316, 1897. doi:10.1080/14786449708621070.

[114] A. Gabrielli, S. Bonacini, K. Kloukinas, A. Marchioro, P. Moreira, A. Ranieri, and D. De Robertis. The GBT-SCA, a radiation tolerant ASIC for detector control applications in SLHC experiments. In *Topical Workshop on Electronics for Particle Physics*, 2009. doi:10.5170/CERN-2009-006.557.

[115] D. Gamba, R. Corsini, and S. Curt et al. The clear user facility at cern. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 909:480–483, 2018. 3rd European Advanced Accelerator Concepts workshop (EAAC2017). doi:10.1016/j.nima.2017.11.080.

[116] E. Georgali, Z. Eleme, N. Patronis, X. Aslanoglou, M. Axiotis, M. Diakaki, V. Foteinou, S. Harissopulos, A. Kalamara, M. Kokkoris, A. Lagoyannis, N. G Nicolis, G. Provatas, A. Stamatopoulos, S. Stoulos, A. Tsinganis, E. Vagena, R. Vlastou, and S. M. Vogiatzi. The $(n, 2n)$ reaction for the lightest stable erbium isotope ^{162}Er from reaction threshold up to 19 mev. *Phys. Rev. C*, 98:014622, Jul 2018. doi:10.1103/PhysRevC.98.014622.

[117] D M Gingrich. Construction, assembly and testing of the ATLAS hadronic end-cap calorimeter. *Journal of Instrumentation*, 2(05):P05005–P05005, may 2007. doi:10.1088/1748-0221/2/05/p05005.

[118] R. Ginosar. Metastability and synchronizers: A tutorial. *IEEE Design Test of Computers*, 28(5):23–35, Sep. 2011. doi:10.1109/MDT.2011.113.

[119] P. Gkountoumis. Level-1 data driver card of the atlas new small wheel upgrade compatible with the phase ii 1 mhz readout scheme. In *2016 5th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, pages 1–4, May 2016. doi:10.1109/MOCAST.2016.7495115.

[120] L. Gonella, F. Faccio, M. Silvestri, S. Gerardin, D. Pantano, V. Re, M. Manghisoni, L. Ratti, and A. Ranieri. Total ionizing dose effects in 130-nm commercial cmos technologies for hep experiments. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 582(3):750–754, 2007. VERTEX 2006. doi:https://doi.org/10.1016/j.nima.2007.07.068.

[121] David J Griffiths. *Introduction to elementary particles; 2nd rev. version*. Physics textbook. Wiley, New York, NY, 2008. URL: https://cds.cern.ch/record/111880.

[122] PCI Special Interest Group. Pci local bus specification. rev. 2.2. Available at https://www.ics.uci.edu/~harris/ics216/pci/PCI_22.pdf, December 1998. Accessed: 15-05-2017.

[123] The NSW Readout Working Group. Requirements for the nsw vmm3 readout asic and the nsw readout controller asic. October 2016. URL: https://espace.cern.ch/ATLAS-NSW-ELX/Shared%20Documents/ROC/VMM3_ROCspec.pdf.

[124] E. Gschwendtner, E. Adli, and L. Amorim et al. Awake, the advanced proton driven plasma wakefield acceleration experiment at cern. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 829:76–82, 2016. 2nd European Advanced Accelerator Concepts Workshop - EAAC 2015. doi:10.1016/j.nima.2016.02.026.

[125] D. Habs, O. Kester, and T. et al. Sieber. The rex-isolde project. *Hyperfine Interactions*, 129(1):43–66, Dec 2000. doi:10.1023/A:1012650908964.

[126] K. HANKE. Past and present operation of the cern ps booster. *International Journal of Modern Physics A*, 28(13):1330019, 2013. doi:10.1142/S0217751X13300196.

[127] H Haseroth, C Hill, and Klaus et al. Langbein. History, developments and recent performance of the cern linac 1. *Proceedings of the 1992 Linear Accelerator Conference, Ottawa, Ontario, Canada*, pages 58–60, 01 1992. URL: https://accelconf.web.cern.ch/l92/papers/mo4-08.pdf.

[128] Werner Herr and B Muratori. Concept of luminosity. 2006. URL: https://cds.cern.ch/record/941318, doi:10.5170/CERN-2006-002.361.

[129] G Iakovidis. The micromegas project for the ATLAS upgrade. *Journal of Instrumentation*, 8(12):C12007–C12007, dec 2013. doi:10.1088/1748-0221/8/12/c12007.

[130] George Iakovidis. VMM3a, an ASIC for tracking detectors. *Journal of Physics: Conference Series*, 1498:012051, apr 2020. doi:10.1088/1742-6596/1498/1/012051.

[131] Georgios Iakovidis. VMM3a, an ASIC for tracking detectors. May 2019. URL: http://cds.cern.ch/record/2675779/files/ATL-MUON-SLIDE-2019-205.pdf?version=1.

[132] Vincenzo Izzo. ATLAS upgrades. *PoS*, LHCP2020:094, 2020. doi:10.22323/1.382.0094.

[133] G.H. Johnson, J.H. Hohl, R.D. Schrimpf, and K.F. Galloway. Simulating single-event burnout of n-channel power mosfet's. *IEEE Transactions on Electron Devices*, 40(5):1001–1008, May 1993. doi:10.1109/16.210211.

[134] A. Kalamara, R. Vlastou, M. Kokkoris, M. Diakaki, A. Tsinganis, N. Patronis, M. Axiotis, and A. Lagoyannis. Investigation of the ^{241}Am$(n,2n)^{240}$Am cross section. *Phys. Rev. C*, 93:014610, Jan 2016. doi:10.1103/PhysRevC.93.014610.

[135] T Kawamoto, S Vlachos, L Pontecorvo, J Dubbert, G Mikenberg, P Iengo, C Dallapiccola, C Amelung, L Levinson, R Richter, and D Lellouch. New Small Wheel Technical Design Report. Technical report, Jun 2013. ATLAS New Small Wheel Technical Design Report. URL: https://cds.cern.ch/record/1552862.

[136] David G. Kendall. Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain. *The Annals of Mathematical Statistics*, 24(3):338 – 354, 1953. doi:10.1214/aoms/1177728975.

[137] S. Kulis. Single event effects mitigation with TMRG tool. *Journal of Instrumentation*, 12(01):C01082–C01082, jan 2017. doi:10.1088/1748-0221/12/01/c01082.

[138] Silicon Labs. Si570/si571. 10 mhz to 1.4 ghz i2c programmable xo/vcxo. Available at https://www.silabs.com/documents/public/data-sheets/si570.pdf. Accessed: 19-02-2017.

[139] C. M. G. Lattes, H. Muirhead, G. P. S. Occhialini, and C. F. Powell. Processes involving charged mesons. *Nature*, 159(4047):694–697, May 1947. doi:10.1038/159694a0.

[140] P Leitao, S Feger, D Porret, S Baron, K Wyllie, M Barros Marin, D Figueiredo, R Francisco, J C Da Silva, T Grassi, and P Moreira. Test bench development for the radiation hard GBTX ASIC. *Journal of Instrumentation*, 10(01):C01038–C01038, jan 2015. doi:10.1088/1748-0221/10/01/c01038.

[141] Lorne Levinson. Nsw electronics overview, Oct 2020. Accessed: 5 January 2021. URL: https://espace.cern.ch/ATLAS-NSW-ELX/Shared%20Documents/Overview%20and%20General/LL_NSW_ElxOvr_notitle_v13.png.

[142] M.E. Levitt. Asic testing upgraded. *IEEE Spectrum*, 29(5):26–29, May 1992. doi:10.1109/6.135405.

[143] Gilbert N. Lewis. The conservation of photons. *Nature*, 118(2981):874–875, Dec 1926. doi:10.1038/118874a0.

[144] A.M. Lombardi. Linac4: From Initial Design to Final Commissioning. In *Proc. of International Particle Accelerator Conference (IPAC'17), Copenhagen, Denmark, 14-19 May, 2017*, number 8 in International Particle Accelerator Conference, pages 1217–1222, Geneva, Switzerland, May 2017. JACoW. doi:10.18429/JACoW-IPAC2017-TUYA1.

[145] N. Madsen. Antiproton physics in the elena era. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2116):20170278, 2018. doi:10.1098/rsta.2017.0278.

[146] Akhilesh Mahajan, Bokka Abhiram Sai Krishna, and Keshava Gopal Goud Cheruku. Logic analyzer for integrated circuits, Jan 2019. US 2019/0026416 A1. URL: https://pdfstore.patentorder.com/pdf/us/416/us2019026416.pdf.

[147] D. Mayhew and V. Krishnan. Pci express and advanced switching: evolutionary path to building next generation interconnects. In *11th Symposium on High Performance Interconnects, 2003. Proceedings.*, pages 21–29, Aug 2003. doi:10.1109/CONECT.2003.1231473.

[148] Stephanie Sammartino McPherson. *Tim Berners-Lee: Inventor of the World Wide Web*. Twenty First Century Books, USA, 1st edition, 2009.

[149] Carver Mead and Lynn Conway. *Introduction to VLSI Systems*. Addison-Wesley, Reading, MA, 1980.

[150] I. P. Mesolongitis, A. Gkountis, E. D. Kyriakis - Bitzaros, K. Zachariadou, P. Gkountoumis, and T. Alexopoulos. Testing the level-1 data driver card for the new small wheel of the atlas detector. In *2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, pages 1–4, May 2017. doi:10.1109/MOCAST.2017.7937667.

[151] P. Miao, F. Li, L. Guan, I. Ravinovich, S. Zhou, N.J. Zhang, Z.L. Zhang, X.X. Wang, and G. Jin. The development of the front-end boards for the small-strip thin gap chambers detector system of the ATLAS muon new small wheel upgrade. *Journal of Instrumentation*, 15(11):P11024–P11024, nov 2020. doi:10.1088/1748-0221/15/11/p11024.

[152] Cush MissMJ. Summary of all known elemental particles in the standard model and their interactions with each other. Available at https://commons.wikimedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg. Accessed: 05-01-2021.

[153] P Moreira, A Marchioro, and Kloukinas. The GBT: A proposed architecure for multi-Gb/s data transmission in high energy physics. 2007. URL: https://cds.cern.ch/record/1091474, doi:10.5170/CERN-2007-007.332.

[154] Inc. Motorola. Spi block guide v03.06. Available at https://web.archive.org/web/20150413003534/http://www.ee.nmt.edu/~teare/ee308l/datasheets/S12SPIV3.pdf, February 2003. Accessed: 15-05-2017.

[155] S. M. Nowick and M. Singh. Asynchronous design—part 1: Overview and recent advances. *IEEE Design Test*, 32(3):5–18, June 2015. doi:10.1109/MDAT.2015.2413759.

[156] S. M. Nowick and M. Singh. Asynchronous design—part 2: Systems and methodologies. *IEEE Design Test*, 32(3):19–28, June 2015. doi:10.1109/MDAT.2015.2413757.

[157] NXP. I2c-bus specification and user manual. um10204. rev. 6. Available at https://www.nxp.com/docs/en/user-guide/UM10204.pdf, April 2014. Accessed: 15-05-2017.

[158] H. Nyquist. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, April 1928. doi:10.1109/T-AIEE.1928.5055024.

[159] Robert Oerter and B. Holstein. The theory of almost everything: The standard model, the unsung triumph of modern physics. *Physics Today - PHYS TODAY*, 59, 07 2006. doi:10.1063/1.2337829.

[160] University of Southern California's Information Sciences Institute. The mosis service. Accessed: 15 June 2020. URL: https://www.mosis.com/learn-more-1.

[161] Preeti Panda, B Silpa, Aviral Shrivastava, and Krishnaiah Gummidipudi. *Power-efficient System Design*. January 2010. doi:10.1007/978-1-4419-6388-8.

[162] Joao Pequenao. Computer Generated image of the ATLAS calorimeter. Mar 2008. URL: https://cds.cern.ch/record/1095927.

[163] Joao Pequenao. Computer generated image of the ATLAS inner detector. Mar 2008. URL: https://cds.cern.ch/record/1095926.

[164] Joao Pequenao. Computer generated image of the ATLAS Muons subsystem. Mar 2008. URL: https://cds.cern.ch/record/1095929.

[165] Joao Pequenao. Computer generated image of the whole ATLAS detector. Mar 2008. URL: https://cds.cern.ch/record/1095924.

[166] Joao Pequenao. Event Cross Section in a computer generated image of the ATLAS detector. Mar 2008. URL: https://cds.cern.ch/record/1096081.

[167] Genessis Perez. *Unitarization Models For Vector Boson Scattering at the LHC*. PhD thesis, 01 2018. doi:10.5445/IR/1000082199.

[168] Max Planck. Ueber das gesetz der energieverteilung im normalspectrum - on the law of distribution of energy in the normal spectrum. *Annalen der Physik*, 4:553–563, 1901. URL: http://web.ihep.su/dbserv/compas/src/planck01/eng.pdf, doi:https://doi.org/10.1002/andp.19013090310.

[169] K Poltorak, F Tavernier, and P Moreira. A radiation-hard PLL for frequency multiplication with programmable input clock and phase-selectable output signals in 130 nm CMOS. *Journal of Instrumentation*, 7(12):C12014–C12014, dec 2012. doi:10.1088/1748-0221/7/12/c12014.

[170] Timothy A. Pontius. Non-power-of-two gray-code counter system having binary incrementer with counts distributed with bilateral symmetry, Sep 2008. EP 1410509B1. URL: https://patentimages.storage.googleapis.com/c7/75/cd/ba5f96171f9013/EP1410509B1.pdf.

[171] S. Popa, M. Ivanovici, and R. M. Coliban. Time-multiplexed 10gbps ethernet-based integrated logic analyzer for fpgas. In *2020 International Symposium on Electronics and Telecommunications (ISETC)*, pages 1–4, Nov 2020. doi:10.1109/ISETC50328.2020.9301115.

[172] S. Popa, M. Luchian, and M. Ivanovici. Clock and data signals synchronization for an fpga-based asic testing setup. In *2019 International Symposium on Signals, Circuits and Systems (ISSCS)*, pages 1–4, July 2019. doi:10.1109/ISSCS.2019.8801780.

[173] S. Popa, S. Mărtoiu, and M. Ivanovici. The quality-control test of the digital logic for the ATLAS new small wheel read-out controller ASIC. *Journal of Instrumentation*, 15(04):P04023–P04023, apr 2020. doi:10.1088/1748-0221/15/04/p04023.

[174] S. Popa, S. Mărtoiu, and M. Ivanovici. Study of the ATLAS new small wheel read-out controller ASIC in a neutron irradiation environment. *Journal of Instrumentation*, 15(10):P10023–P10023, oct 2020. doi:10.1088/1748-0221/15/10/p10023.

[175] Stefan Popa, Sorin Martoiu, Mihai Luchian, Radu Coliban, and Mihai Ivanovici. The Quality-Assurance Test of the ATLAS New Small Wheel Read-Out Controller ASIC. In *Proceedings of Topical Workshop on Electronics for Particle Physics — PoS(TWEPP2018)*, volume 343, page 081, 2019. doi:10.22323/1.343.0081.

[176] J. Postel. Internet protocol. Available at https://doi.org/10.17487/rfc0791, September 1981.

[177] ATLAS Electronics Coordination TDAQ Phase-II Upgrade Project. Atlas trigger & daq interfaces with detector front-end systems: Requirement document for hl-lhc. Atlas doc.: At2-di-es-0002, edms id: 1563801 v.1, CERN, Oct 2018. URL: https://edms.cern.ch/ui/file/1563801/1/RequirementsPhaseII_v1.1.0.pdf.

[178] Fabrizio Scuri. Upgrade of the ATLAS tile calorimeter for the high luminosity LHC. *Journal of Physics: Conference Series*, 1162:012017, jan 2019. doi:10.1088/1742-6596/1162/1/012017.

[179] Abraham Seiden. Characteristics of the atlas and cms detectors. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1961):892–906, 2012. doi:10.1098/rsta.2011.0461.

[180] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, Jan 1949. doi:10.1109/JRPROC.1949.232969.

[181] Martin Shooman. *N-Modular Redundancy*, pages 145–201. 03 2002. doi:10.1002/047122460X.ch4.

[182] Inc. Silicon Laboratories. Selecting the optimum pci express clock source. rev. 2.0. Available at https://www.silabs.com/documents/public/white-papers/PCIe-Clock-Source-Selection.pdf. Accessed: 15-05-2017.

[183] Tejinder Singh, Farzaneh Pashaie, and Rajat Kumar. Redundancy based design and analysis of alu circuit using cmos 180nm process technology for fault tolerant computing architectures. *International Journal of Computing and Digital Systems*, 4:53–62, 01 2015. doi:10.12785/ijcds/040106.

[184] Helmuth Spieler. *Analog and Digital Electronics for Detectors*. Physics Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, U.S.A. URL: https://www-physics.lbl.gov/~spieler/ICFA_Rio_2003/text/Analog_and_Digital_Electronics_for_Detectors.pdf.

[185] Bernd Stelzer. The new small wheel upgrade project of the atlas experiment. *Nuclear and Particle Physics Proceedings*, 273-275:1160–1165, 2016. 37th International Conference on High Energy Physics (ICHEP). doi:10.1016/j.nuclphysbps.2015.09.182.

[186] Jean-Philippe et al. Tock. The Second LHC Long Shutdown (LS2) for the Superconducting Magnets. In *9th International Particle Accelerator Conference*, 6 2018. doi:10.18429/JACoW-IPAC2018-MOPMF056.

[187] Claudio Leopoldo Torregrosa Martin. Comprehensive Study for an Optimized Redesign of the CERN's Antiproton Decelerator Target, Apr 2018. Presented 12 Mar 2018. URL: https://cds.cern.ch/record/2314375.

[188] Fang-Ying Tsai. Searching for Heavy Higgs Bosons Using Events with Leptons and Missing Transverse Momentum with the ATLAS Detector, Sep 2020. Presented 26 Oct 2020. Accessed 5 January 2021. URL: https://cds.cern.ch/record/2743674.

[189] P. Tzanis. Electronics performance of the ATLAS new small wheel micromegas wedges at CERN. *Journal of Instrumentation*, 15(07):C07002–C07002, jul 2020. doi:10.1088/1748-0221/15/07/c07002.

[190] Theodoros Vafeiadis. Integration and commissioning of ATLAS New Small Wheel Micromegas detectors with electronics at CERN. In *Proceedings of 40th International Conference on High Energy physics — PoS(ICHEP2020)*, volume 390, page 791, 2 2021. doi:10.22323/1.390.0791.

[191] Peter Vankov and ATLAS Collaboration. ATLAS Future Upgrade. Technical report, CERN, Geneva, Jun 2016. URL: https://cds.cern.ch/record/2195333, doi:10.22323/1.273.0061.

[192] T. S. Virdee. Beyond the standard model of particle physics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2075):20150259, 2016. doi:10.1098/rsta.2015.0259.

[193] R. Vlastou, M. Anastasiou, A. Kalamara, M. Diakaki, M. Kokkoris, V. Paneta, M. Axiotis, A. Lagoyannis, N. Nicolis, and M. Serris. High-energy neutron facility at the athens tandem accelerator ncsr "demokritos". *HNPS Advances in Nuclear Physics*, 21(0):123–126, 2019. doi:10.12681/hnps.2015.

[194] R. Vlastou, D. Sigalas, A. Kalamara, M. Kokkoris, M. Anastasiou, M. Diakaki, M. Axiotis, and A. Lagoyannis. Neutron beam characterization at the athens tandem accelerator ncsr "demokritos". *HNPS Advances in Nuclear Physics*, 23(0):34–38, 2019. doi:10.12681/hnps.1904.

[195] A Vogel. ATLAS Transition Radiation Tracker (TRT): Straw Tube Gaseous Detectors at High Rates. Technical report, CERN, Geneva, Apr 2013. URL: http://cds.cern.ch/record/1537991.

[196] Jinhong Wang, Liang Guan, J. W. Chapman, Bing Zhou, and Junjie Zhu. Design of a trigger data serializer asic for the upgrade of the atlas forward muon spectrometer. *IEEE Transactions on Nuclear Science*, 64(12):2958–2965, Dec 2017. doi:10.1109/TNS.2017.2771266.

[197] Xu Wang. Frontend and backend electronics for the New Small Wheel Upgrade of the ATLAS muon spectrometer, 2018. Accessed: 21 September 2018. URL: https://indico.cern.ch/event/697988/contributions/3055968/attachments/1719124/2774476/Frontend_and_backend_electronics_for_the_New_Small_Wheel_Upgrade_of_the_ATLAS_muon_spectrometer.pdf.

[198] Jorg Wenninger. Operation and Configuration of the LHC in Run 2. Mar 2019. URL: https://cds.cern.ch/record/2668326.

[199] A. X. Widmer and P. A. Franaszek. A dc-balanced, partitioned-block, 8b/10b transmission code. *IBM Journal of Research and Development*, 27(5):440–451, Sep. 1983. doi:10.1147/rd.275.0440.

[200] J Wotschack. ATLAS Muon Chamber Construction Parameters for CSC, MDT, and RPC chambers. Technical report, CERN, Geneva, Apr 2008. Back-up document for the ATLAS Detector Paper. URL: http://cds.cern.ch/record/1099400.

[201] K. Wyllie, S. Baron, S. Bonacini, Ö. Çobanoğlu, F. Faccio, S. Feger, R. Francisco, P. Gui, J. Li, A. Marchioro, P. Moreira, C. Paillard, and D. Porret. A gigabit transceiver for data transmission in future high energy physics experiments. *Physics Procedia*, 37:1561–1568, 2012. Proceedings of the 2nd International Conference on Technology and Instrumentation in Particle Physics (TIPP 2011). doi:https://doi.org/10.1016/j.phpro.2012.02.487.

[202] Xilinx. Ultrascale architecture clocking resources. user guide. ug572 (v1.10). Available at https://www.xilinx.com/support/documentation/user_guides/ug572-ultrascale-clocking.pdf. Accessed: 19-02-2017.

[203] Xilinx. Ultrascale architecture selectio resources. user guide. ug571 (v1.12). Available at https://www.xilinx.com/support/documentation/user_guides/ug571-ultrascale-selectio.pdf. Accessed: 19-02-2020.

[204] Xilinx. Axi iic bus interface v2.0 logicore ip product guide. pg090. Available at https://www.xilinx.com/support/documentation/ip_documentation/axi_iic/v2_0/pg090-axi-iic.pdf, October 2016. Accessed: 15-03-2018.

[205] Xilinx. Axi timer v2.0 logicore ip product guide. pg079. Available at https://www.xilinx.com/support/documentation/ip_documentation/axi_timer/v2_0/pg079-axi-timer.pdf, October 2016. Accessed: 15-03-2018.

[206] Xilinx. Integrated Logic Analyzer v6.2, PG172. Available at https://www.xilinx.com/support/documentation/ip_documentation/ila/v6_2/pg172-ila.pdf, 2016. Accessed 14-05-2020.

[207] Xilinx. Axi interconnect v2.1 logicore ip product guide. pg059. Available at https://www.xilinx.com/support/documentation/ip_documentation/axi_interconnect/v2_1/pg059-axi-interconnect.pdf, December 2017. Accessed: 15-03-2018.

[208] Xilinx. Axi uart lite v2.0 logicore ip product guide. pg142. Available at https://www.xilinx.com/support/documentation/ip_documentation/axi_uartlite/v2_0/pg142-axi-uartlite.pdf, April 2017. Accessed: 15-03-2018.

[209] Xilinx. Kcu105 board, ug917 (v1.10). Available at https://www.xilinx.com/support/documentation/boards_and_kits/kcu105/ug917-kcu105-eval-bd.pdf, February 2019. Accessed: 19-04-2021.

[210] Xilinx. Ultrascale architecture. gth transceivers. user guide. ug576. Available at https://www.xilinx.com/support/documentation/user_guides/ug576-ultrascale-gth-transceivers.pdf, August 2019. Accessed: 19-04-2021.

[211] Xilinx. Kintex ultrascale fpgas data sheet: Dc and ac switching characteristics. ds892 (v1.19). Available at https://www.xilinx.com/support/documentation/data_sheets/ds892-kintex-ultrascale-data-sheet.pdf, September 2020. Accessed: 19-02-2021.

[212] Xilinx. Microblaze processor reference guide. ug984 (v2020.2). Available at https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_2/ug984-vivado-microblaze-ref.pdf, November 2020. Accessed: 19-02-2021.

[213] Xilinx. Microblaze debug module v3.2 logicore ip product guide. pg115 (v3.2). Available at https://www.xilinx.com/support/documentation/ip_documentation/mdm/v3_2/pg115-mdm.pdf, January 2021. Accessed: 19-02-2021.

[214] Xilinx. UltraScale Architecture and Product Data Sheet: Overview, DS890. Available at https://www.xilinx.com/support/documentation/data_sheets/ds890-ultrascale-overview.pdf, March 2021. Accessed: 19-04-2021.

[215] S. Yano, K. Nakamura, T. Bitoh, Y. Konno, and K. Saga. A consistent scan design system for large-scale asics. In *2012 IEEE 21st Asian Test Symposium*, page 82, Los Alamitos, CA, USA, nov 1996. IEEE Computer Society. doi:10.1109/ATS.1996.555141.

[216] L. Yao, H. Chen, K. Chen, S. Tang, and V. Polychronakos. Design and Testing of the Address in Real-Time Data Driver Card for the Micromegas Detector of the ATLAS New Small Wheel Upgrade. *IEEE Trans. Nucl. Sci.*, 67(9):2155–2160, 2020. arXiv:1806.06912, doi:10.1109/TNS.2020.2986418.

[217] L. Yao, V. Polychronakos, H. Chen, K. Chen, H. Xu, S. Martoiu, N. Felt, and T. Lazovich. The address in real time data driver card for the MicroMegas detector of the ATLAS muon upgrade. *Journal of Instrumentation*, 12(01):C01047–C01047, jan 2017. doi:10.1088/1748-0221/12/01/c01047.

[218] Hideki Yukawa. On the interaction of elementary particles. i. *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, 17:48–57, 1935. doi:10.11429/ppmsj1919.17.0_48.

[219] Omid Zeynali, Daryoush Masti, Maryam Nezafat, and Alireza Mallahzadeh. Study of "radiation effects of nuclear high energy particles" on electronic circuits and methods to reduce its destructive effects. *Journal of Modern Physics*, 2(12):1567–1573, 2011. doi:10.4236/jmp.2011.212191.